

Projecte de Fi de Carrera

Lídia Alcobero Gabarró

Sistema d'interacció implícita en realitat virtual, amb control per radiofreqüència

ANNEX A: Recull de les especificacions acordades per al projecte

ANNEX B: Elecció de components dels mòduls actuadors

ANNEX C: Elecció de components de la unitat central

ANNEX D: Elecció de components del mòdul de programació

ANNEX E: Impacte ambiental

ANNEX F: Pressupost i termini

ANNEX G: Càlcul de l'abast

ANNEX H: Esquemàtics dels prototipus finals

ANNEX I: Codi implementat en la unitat central

ANNEX J: Codi implementat en el node actuador d'un vibrador

ANNEX K: Codi implementat en el node actuador de 3 vibradors

ANNEX L: Codi de la llibreria de control RFControlAPI

ANNEX M: Implementació de la interfície gràfica RFControl

Autor: Lídia Alcobero Gabarró

Director: Emili Lupón Rosés

Convocatoria: Febrer 2005 (pla 94)



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

En l'**Annex A** es recullen les **especificacions finals** del sistema tal i com es van definir i van ser acordades per l'Emili Lupón Rosés conjuntament amb SEAT a l'inici del projecte.

En els **Annexos B, C i D** es justifica respectivament l'**elecció dels components** dels mòduls actuadors de la unitat central i del mòdul de programació de l'EEPROM.

L'**Annex E** consisteix en l'estudi de l'**impacte ambiental** del projecte, prenent en consideració l'impacte causat per la fabricació dels circuits, les emissions en temps d'ús i l'impacte de les bateries i de la transmissió per radiofreqüència, així com el final de la vida útil dels components.

L'**Annex F** presenta el **pressupost total** del sistema, així com les **referències comercials** de cadascun dels components que formen el sistema final per al client i l'entorn de disseny, els seus distribuïdors i el seu cost. També s'inclou el primer **estudi de termini** que es va realitzar a l'inici del projecte.

En l'**Annex G** és on s'han inclòs unes **nocions bàsiques** sobre radiofreqüència i la determinació teòrica de l'**abast màxim** que tindria el sistema dissenyat.

L'**Annex H** recull els **esquemàtics** dels prototipus finals del sistema, així com dels elements de recolzament dissenyats (el mòdul de programació) i d'altres elements com per exemple la placa de proves.

Els darrers annexos inclouen el **codi font implementat** en els diferents mòduls. En l'**Annex I** el de la unitat central, l'**Annex J** el dels nodes actuadors d'un vibrador i l'**Annex K** el dels nodes de 3 vibradors.

Pel què respecta a la comunicació mitjançant el computador, en l'**Annex L** hi consta el codi de la **llibreria de control** del sistema, i finalment en l'**Annex M** el de la implementació de la **interfície gràfica**.

Sumari

RESUM	3
SUMARI	5
A. RECULL DE LES ESPECIFICACIONS ACORDADES PER AL PROJECTE	7
A.1. Estructura general del sistema	7
A.2. El transceptor de radiofreqüència i el microcontrolador	10
A.3. L'enllaç de radiofreqüència	11
A.4. La unitat central de radiofreqüència.....	13
A.5. Els nodes de vibradors.....	15
A.6. La interfície en el PC	18
B. ELECCIÓ DE COMPONENTS DELS MÒDULS ACTUADORS	19
B.1. Elecció del mòdul de radiofreqüència	19
B.2. Elecció del motor vibrador.....	23
B.3. Elecció de la bateria.....	24
B.4. Elecció de l'EEPROM	26
B.5. Elecció d'altres components	27
B.6. Elecció de l'encapsulat.....	33
C. ELECCIÓ DELS COMPONENTS DE LA UNITAT CENTRAL	35
D. ELECCIÓ DELS COMPONENTS DEL MÒDUL DE PROGRAMACIÓ DE L'EEPROM	39
E. IMPACTE AMBIENTAL	41
E.1. Impacte de la fabricació dels circuits integrats i impresos	41
E.2. Emissions químiques degudes a la utilització dels components electrònics.....	42
E.3. Impacte ambiental de les bateries dels nodes actuadors	43
E.4. Final de la vida dels components.....	45
E.5. Impacte degut a la transmissió per radiofreqüència.....	47

F. PRESSUPOST I TERMINI	49
F.1. Pressupost	49
F.1.1. Pressupost de la unitat central	50
F.1.2. Pressupost de les unitats actuadores d'un vibrador	52
F.1.3. Pressupost de les unitats actuadores de tres vibradors	54
F.1.4. Pressupost del mòdul programador de l'EEPROM	56
F.1.5. Pressupost del hardware de desenvolupament i de proves	57
F.1.6 Càlcul del cost total	59
F.2. Terminis	62
G. CÀLCUL DE L'ABAST	65
G.1. Nocions bàsiques sobre radiofreqüència	65
G.2. Determinació de l'abast màxim	70
H. ESQUEMÀTICS DELS PROTOTIPUS FINALS	75
H.1. Esquemàtic de la unitat central	75
H.2. Esquemàtic dels nodes actuadors d'un vibrador	76
H.3. Esquemàtic dels nodes actuadors de tres vibradors	77
H.4. Esquemàtic del mòdul de programació de l'EEPROM	78
H.5. Esquemàtic de la placa de proves	79
I. CODI IMPLEMENTAT EN LA UNITAT CENTRAL	81
J. CODI IMPLEMENTAT EN EL NODE ACTUADOR D'UN VIBRADOR	91
K. CODI IMPLEMENTAT EN ELS NODES DE 3 VIBRADORS	107
L. CODI DE LA LLIBRERIA DE CONTROL RFCONTROLAPI	127
M. IMPLEMENTACIÓ DE LA INTERFÍCIE GRÀFICA RFCONTROL	133
M.1. Implementació de la pantalla principal	133
M.2. Implementació de la pantalla d'estat global de bateria	139

A. Recull de les especificacions acordades per al projecte

En aquest annex es recullen les especificacions finals del sistema tal i com s'han definit. Han estat acordades per l'Emili Lupón Rosés conjuntament amb SEAT, però això no exclou que el mateix sistema pugui ser utilitzat per a altres entorns i aplicacions. El projecte final aconsegueix amb totes aquestes especificacions excepte pel que respecta al canal de transmissió, el qual s'ha variat perquè s'han trobat opcions més adequades. S'han introduït també millores i noves funcionalitats no especificades en l'acord.

ESPECIFICACIONES FUNCIONALES DEL SISTEMA DE VIBRADORES ELECTROMECAÑICOS CONTROLADOS DESDE UN PC VÍA UNA CONEXIÓN INALÁMBRICA

A.1. Estructura general del sistema

El sistema de vibradores electromecánicos controlados desde un PC vía una conexión inalámbrica estará integrado por un conjunto de hasta treinta y un vibradores electromecánicos cuyo estado de vibración se podrá gobernar a distancia desde un PC.

Cada vibrador dispondrá de un identificador propio de cinco bits que le permitirá distinguirse de los otros vibradores del sistema.

Los vibradores utilizados serán similares a los existentes en los teléfonos móviles, que se basan en un pequeño motor con una carga mecánica excéntrica. Dado que dichos vibradores no pueden estar operativos continuamente por un problema de disipación térmica, la vibración consistirá en una excitación periódica de frecuencia f con un ciclo de trabajo c apropiado (entre el 25% y el 70%). La frecuencia de la excitación f y el ciclo de trabajo c definirán las características de la vibración.

Los vibradores admitirán, además del estado de reposo (no vibración), hasta tres tipos de vibración diferentes, que se caracterizarán por distintos valores de la frecuencia de la excitación f y del ciclo de trabajo c .

El estado de vibración de cada uno de los vibradores será independiente del estado de vibración de los otros vibradores y podrá seleccionarse de entre las cuatro alternativas indicadas en el párrafo anterior. El estado de vibración de cada uno de los vibradores se

codificará mediante dos bits, de acuerdo a los siguientes códigos y características de vibración:

- 00, vibrador en reposo (no vibración).
- 01, vibración de frecuencia 2 Hz y ciclo de trabajo del 20% (100 ms vibrando cada 500 ms).
- 10, vibración de frecuencia 5 Hz y ciclo de trabajo del 50% (100 ms vibrando cada 200 ms).
- 11, vibración de frecuencia 25 Hz y ciclo de trabajo del 50% (20 ms vibrando cada 40 ms).

Serán, pues, necesarios sesenta y dos bits para codificar el estado de todos los vibradores del sistema.

Los distintos vibradores del sistema estarán distribuidos en nodos de vibradores, considerándose únicamente los siguientes dos tipos de nodos:

- Nodos con un único vibrador.
- Nodos con tres vibradores independientes, destinados originariamente a los dedos pulgar, índice y corazón de una mano. Los identificadores de los vibradores de estos nodos presentarán valores consecutivos, el menor correspondiendo al vibrador del dedo pulgar, el intermedio al vibrador del dedo índice y el mayor al vibrador del dedo corazón.

Cada uno de estos nodos de vibradores estará completamente aislado, desde un punto de vista eléctrico, del resto del sistema. En consecuencia, no debe existir ningún tipo de cableado entre un nodo de vibradores y los restantes elementos integrantes del sistema (otros nodos de vibradores, PC controlador, etc.), tanto en lo que se refiere a la indicación del estado de los vibradores del nodo como a la obtención de la energía eléctrica necesaria para su funcionamiento. Para conseguir esta prestación, es necesario satisfacer las siguientes condiciones:

- Cada nodo de vibradores ha de ser energéticamente autónomo, alimentándose de una batería propia que forme parte integrante del nodo.
- El estado de los vibradores del sistema debe ser indicado a los nodos de vibradores a través de una conexión inalámbrica.

La transmisión inalámbrica de información puede realizarse utilizando infrarrojos, ultrasonidos o radiofrecuencia. La existencia, dentro del espectro de radiofrecuencia, de una serie de bandas libres para su utilización en aplicaciones industriales, científicas y médicas (bandas ISM), la reciente aparición de una gran variedad de dispositivos electrónicos que facilitan la comunicación utilizando estas bandas y la distancia y ausencia

de visión directa entre los elementos que deben comunicarse (PC y elementos vibradores), hacen de la transmisión inalámbrica por radiofrecuencia la más apropiada para ser utilizada en esta aplicación.

En consecuencia, el sistema de vibradores electromecánicos controlados desde un PC vía una conexión inalámbrica estará integrado por los siguientes elementos:

- Un conjunto de hasta treinta y un vibradores electromecánicos, independientes entre si, que estarán distribuidos en nodos de vibradores. Existirán dos tipos distintos de nodos de vibradores con características constructivas diferentes, uno soportando un único vibrador y otro soportando tres vibradores destinados a los dedos pulgar, índice y corazón de una mano. Cada nodo de vibradores deberá disponer de una batería y de un transceptor de radiofrecuencia propios, así como de un microcontrolador que interprete la información transmitida mediante radiofrecuencia y gestione la operación de vibración de los vibradores de acuerdo a dicha información.
- Un PC, que determinará y ordenará el estado de los vibradores electromecánicos a partir de un software de control al efecto.
- Un elemento que, conectado al PC, permita a éste la transmisión del estado de los vibradores a los nodos de vibradores utilizando radiofrecuencia. Este elemento se denominará unidad central de radiofrecuencia y deberá disponer de una fuente de alimentación derivada de la red eléctrica y de un transceptor de radiofrecuencia propios, así como de un microcontrolador que interprete los comandos provenientes del PC y gestione la transmisión de información mediante radiofrecuencia.

El sistema de vibradores electromecánicos controlados desde un PC vía una conexión inalámbrica dispondrá, a fin de facilitar las tareas de su mantenimiento, de un procedimiento que permita modificar los identificadores de los vibradores presentes en el sistema desde el PC. De este modo, un nodo de vibradores de recambio podrá servir para sustituir a cualquier nodo de vibradores del mismo tipo que se haya averiado, sean cuales sean los identificadores de los vibradores de dicho nodo (uno o tres), sin tener que manipular su interior. Durante el procedimiento de modificación de los identificadores de los vibradores sólo se deberá tener en funcionamiento aquel nodo cuyos identificadores de vibradores se deseen modificar, pudiéndose modificar los identificadores de los vibradores de varios nodos nodo tras nodo.

En los próximos apartados se detallarán las características funcionales de todos estos elementos. En primer lugar, en el apartado 2, se describirán el transceptor de radiofrecuencia y el microcontrolador elegidos como dispositivos fundamentales de la unidad central de radiofrecuencia y de los nodos de vibradores. Seguidamente, en el apartado 3, se detallarán las características principales del enlace de radiofrecuencia establecido entre la unidad central de radiofrecuencia y los nodos de vibradores, incluido el formato de las tramas transmitidas mediante radiofrecuencia. A continuación, en los

apartados 4 y 5, se describirán, respectivamente, la unidad central de radiofrecuencia, incluyendo el formato de las tramas transmitidas desde y hacia el PC, y los nodos de vibradores. Finalmente, en el apartado 6, se detallará la interfaz en el PC.

A.2. El transceptor de radiofreqüència i el microcontrolador

El enlace por radiofrecuencia existente entre la unidad central de radiofrecuencia y los distintos nodos de vibradores estará soportado por transceptores (emisores/receptores) de radiofrecuencia operando con unas mismas características: misma banda ISM, mismo canal (misma frecuencia de la portadora), mismo sistema de modulación de la información, mismo encabezamiento de las tramas transmitidas, misma tipo de redundancia en las tramas (CRC), etc. En consecuencia, es conveniente utilizar el mismo tipo de transceptor en todos los elementos mencionados, ya que así se garantiza que todos ellos podrán operar de la misma manera y, por consiguiente, serán compatibles.

Asimismo, tanto la unidad central de radiofrecuencia como los distintos nodos de vibradores estarán materializados alrededor de un microcontrolador cuya misión será la de gestionar todas las entradas y salidas del elemento, incluido el transceptor de radiofrecuencia. Evidentemente, también en este caso es conveniente utilizar el mismo tipo de microcontrolador, ya que de esta forma se minimizan los esfuerzos materiales y humanos necesarios para desarrollar tanto el hardware como el software de las distintas aplicaciones.

Teniendo presente la necesidad de utilizar un transceptor de radiofrecuencia y un microcontrolador, cabe considerar dos alternativas:

- Utilizar dos componentes distintos e independientes, es decir, uno que sea un microcontrolador y otro que sea un transceptor de radiofrecuencia. Esta alternativa abre las puertas a un gran número de posibilidades.
- Utilizar un único componente, es decir, un microcontrolador que incluya un transceptor de radiofrecuencia entre los periféricos integrados en él. Esta solución tiene la ventaja de minimizar el número de componentes necesarios para implementar los distintos elementos del sistema y, presumiblemente, el coste, tamaño y consumo de dichos elementos. En su detrimento cabe indicar el reducido número de componentes con estas prestaciones existentes hoy en día en el mercado.

Hechas estas consideraciones, y tras una exhaustiva investigación del mercado, se ha optado por la segunda de las alternativas, eligiendo como componente fundamental de la unidad central de radiofrecuencia y de los distintos nodos de vibradores el dispositivo nRF9E5 de Nordic, aparecido recientemente en el mercado (febrero del 2004),

implementado con las tecnologías más modernas y dotado de las mejores prestaciones en lo que al transceptor de radiofrecuencia se refiere.

Este dispositivo es un microcontrolador con un microprocesador compatible con el 8051 de Intel que incluye un transceptor de radiofrecuencia capaz de trabajar en las bandas ISM de 433 MHz (Europa y EEUU), 868 MHz (Europa) y 915 MHz (EEUU). Sólo requiere de una tensión de alimentación comprendida en el rango de 1,9 V a 3,6 V, por lo que puede alimentarse perfectamente con una pila de litio de tensión nominal 3 V y tensión final de descarga 2 V. Su capacidad de direccionamiento es de 4 kbytes de memoria de programa/datos más 256 bytes de memoria de datos. Su subsistema de entrada/salida incluye, además del transceptor de radiofrecuencia, 12 pines bidireccionales de entrada/salida para señales digitales, de los que 4 pueden suministrar corrientes elevadas limitadas a 10 mA y otros 4 pueden absorber corrientes elevadas también limitadas a 10 mA, un convertidor A/D de 10 bits con 4 canales analógicos de entrada y velocidad de conversión de 80.000 muestras por segundo, una UART, un bus SPI y varios temporizadores. Su encapsulado es muy pequeño, del tipo QFN de 32 pines (5x5 mm). Además, presenta muchos modos operativos de bajo consumo y permite dejar temporalmente en stand-by, es decir, sin consumo apreciable, aquellos periféricos de entrada/salida (incluido el transceptor de radiofrecuencia) no utilizados en cada momento, por lo que resulta ideal para ser utilizado en sistemas alimentados por baterías que requieran una elevada autonomía.

Otras ventajas de la utilización del nRF9E5 de Nordic son la exhaustiva documentación sobre la utilización del dispositivo y sobre la problemática de los enlaces de radiofrecuencia disponible en la página web del fabricante, la posibilidad de obtener software libre para el desarrollo de aplicaciones basadas en el microprocesador 8051 y la existencia de un kit de evaluación del dispositivo.

A.3. L'enllaç de radiofreqüència

De acuerdo a las características y posibilidades del transceptor de radiofrecuencia incorporado en el dispositivo nRF9E5 de Nordic, el enlace de radiofrecuencia utilizado en el sistema se establecerá en el canal de la banda ISM de 868MHz caracterizado por una frecuencia central de 868,4MHz y un ancho de banda de 100kHz.

El tipo de modulación empleado en la transmisión será GFSK (Gaussian Frequency Shift Keying), con una velocidad de transmisión de 100kbits por segundo.

Para la transmisión de los datos se utilizará codificación Manchester, codificación redundante que utiliza una secuencia de dos bits por cada bit de datos transmitido

(secuencias 10 o 01), por lo que la velocidad efectiva de transmisión será sólo de 50 kbits de datos por segundo. La utilización de este tipo de codificación se utiliza en muchos transceptores de radiofrecuencia ya que permite aumentar su sensibilidad (facilidad de discriminación de los 0's y 1's transmitidos en función del nivel de la señal recibida), la cual depende de que existan muchas transiciones entre los valores 0 y 1, garantizando el código Manchester al menos una transición por cada bit de datos transmitido.

Las tramas transmitidas por el enlace de radiofrecuencia comprenderán, además del correspondiente preámbulo añadido por el transceptor, los siguientes bytes de datos en el orden en que se indican:

- 4 bytes correspondientes a la dirección de la red, que será la dirección hexadecimal DEE6EDA5.
- 8 bytes con la información para los nodos de vibradores o bien un único byte con la información para la unidad central de radiofrecuencia, informaciones que serán detalladas más adelante en este mismo apartado.
- 2 bytes redundantes de CRC (Cyclic Redundancy Check).

Toda trama cuya dirección de la red no coincida con la dirección arriba indicada o cuyo CRC sea incorrecto será descartada.

La duración de la transmisión de una trama emitida desde la unidad central de radiofrecuencia será de unos de 2,89 ms (14×8 bits a $20 \mu\text{s/bit}$ más $650 \mu\text{s}$ necesarios para la generación y/o detección del preámbulo).

La duración de la transmisión de una trama emitida desde un nodo de vibradores será de unos 1,77 ms (7×8 bits a $20 \mu\text{s/bit}$ más $650 \mu\text{s}$ necesarios para la generación y/o detección del preámbulo).

La información para los nodos de vibradores, siempre integrada por 8 bytes, podrá ser de dos tipos:

- Definición del estado de vibración de los treinta y un vibradores electromecánicos del sistema. Cada uno de los 8 bytes se dividirá en cuatro campos de dos bits. Los bits 1 y 0 integrarán el primer campo, los bits 3 y 2 el segundo, los bits 5 y 4 el tercero y los bits 7 y 6 el cuarto. El primer campo del primer byte presentará el valor binario 00. Cada uno de los restantes campos codificará el estado de vibración de uno de los treinta y un vibradores, de acuerdo a la siguiente ordenación: el campo i ($1 \leq i \leq 4$) del byte j ($1 \leq j \leq 8$) codificará el estado del vibrador con identificador $4 \times j + i - 5$ (identificador entre 1 y 31).
- Indicación de los identificadores de los vibradores de los nodos activos del sistema. Los 8 bytes tendrán idéntico contenido, lo que aumentará la fiabilidad de esta trama que contiene información crítica y que sólo se transmite puntualmente. Cada byte

se dividirá en dos campos. Los bits 2 a 0 integrarán el primer campo y los bits 7 a 3 integrarán el segundo. El primer campo presentará el valor binario 011, mientras que el segundo indicará el menor de los identificadores de los varios (uno o tres) que se deben asignar a los vibradores del único nodo de vibradores que debiera estar en funcionamiento.

La información para la unidad central de radiofrecuencia, integrada por un sólo byte, sólo podrá ser de un tipo y reconocerá la correcta recepción por un nodo de vibradores de una trama de indicación de los identificadores de los vibradores de los nodos activos del sistema. El contenido de dicho byte coincidirá con el de los bytes de información de la trama reconocida.

A.4. La unitat central de radiofreqüència

La unidad central de radiofrecuencia es aquel elemento que, conectado al PC, ha de facilitar a éste la capacidad de transmitir el estado de los vibradores utilizando radiofrecuencia.

La conexión entre el PC y la unidad central de radiofrecuencia se realizará mediante un enlace serie que seguirá el estándar RS232C (puerto serie del PC), trabajando a 19200 baudios y transmitiendo caracteres con 8 bits de datos, paridad par y un bit de stop. El cable de conexión podrá tener cualquier longitud soportada por el estándar a la velocidad de transmisión indicada.

A través del enlace serie descrito en el párrafo anterior, se enviarán comandos desde el PC a la unidad central de radiofrecuencia, así como reconocimientos del cumplimiento de estos comandos desde la unidad central de radiofrecuencia hacia el PC. Todos los comandos y reconocimientos estarán constituidos por un único carácter.

Los comandos enviados desde el PC a la unidad central de radiofrecuencia podrán ser de dos tipos:

- Definición del nuevo estado de vibración de un vibrador. Los 8 bits del comando se dividirán en tres campos. El bit 0 integrará el primer campo, los bits 2 y 1 el segundo y los bits 7 a 3 el tercero. El primer campo presentará el valor binario 0, mientras que el segundo y el tercero indicarán, respectivamente, el nuevo estado de vibración del vibrador y el identificador del vibrador.
- Indicación de los identificadores de los vibradores de los nodos activos del sistema. Los 8 bits del comando se dividirán en dos campos. Los bits 2 a 0 integrarán el primer campo y los bits 7 a 3 el segundo. El primer campo presentará el valor binario 011, mientras que el segundo indicará el menor de los identificadores de los varios (uno o tres) que se deben asignar a los vibradores del único nodo de vibradores que debiera estar en funcionamiento.

Todo comando enviado desde el PC a la unidad central de radiofrecuencia tendrá asociado una respuesta de reconocimiento, cuyo formato será idéntico al del comando (eco), y que se transmitirá una vez ejecutado el comando por parte de la unidad central de radiofrecuencia.

La unidad central de radiofrecuencia se basará en el microcontrolador nRF9E5 de Nordic, que servirá de puente entre el enlace serie RS232C proveniente del PC gestionado mediante la UART integrada en el microcontrolador y el transceptor de radiofrecuencia también integrada en el microcontrolador.

A partir de los comandos recibidos desde el PC, la unidad central de radiofrecuencia establecerá y memorizará el estado de vibración de todos los vibradores del sistema, el cual transmitirá a todos los elementos vibradores del sistema mediante una trama "broadcast" de radiofrecuencia. Esta trama se repetirá cada 100 ms con la información actualizada de acuerdo a los comandos recibidos desde el PC.

Esta repetición periódica y frecuente de la trama "broadcast" elude la necesidad de utilizar tramas de radiofrecuencia para reconocer la correcta recepción de la trama "broadcast", tramas que se utilizan en los protocolos de comunicación entre sistemas caracterizados por la transmisión de tramas puntuales en el tiempo que deben ser retransmitidas hasta haberse asegurado de su correcta recepción.

La repetición periódica, a intervalos conocidos, de la trama "broadcast" y la ausencia de tramas de reconocimiento de dicha trama posibilitarán un ahorro significativo del consumo de los nodos de vibradores, por lo que estos ganarán en autonomía.

La unidad central de radiofrecuencia enviará al PC el reconocimiento de un comando de definición del nuevo estado de vibración de un vibrador una vez haya actualizado su registro del estado de vibración de todos los vibradores del sistema.

Cuando la unidad central de radiofrecuencia reciba desde el PC un comando de indicación de los identificadores de los vibradores de los nodos activos del sistema, generará una trama del mismo tipo y la transmitirá por radiofrecuencia hacia el único nodo de vibradores del sistema que debiera estar en funcionamiento. Acto seguido, esperará la trama de reconocimiento correspondiente que debiera ser transmitida por dicho nodo y, tras recibirla correctamente, enviará el correspondiente reconocimiento hacia el PC. Si tras una espera de 10 ms no llegara la trama de reconocimiento transmitida por el nodo o bien llegara una trama errónea, la unidad central de radiofrecuencia retransmitirá por radiofrecuencia la trama de indicación de los identificadores de los vibradores de los nodos activos del sistema y procederá a una nueva espera de la correspondiente trama de reconocimiento,

repetiéndose indefinidamente el procedimiento antes descrito hasta la recepción de una trama correcta de reconocimiento.

La unidad central de radiofrecuencia se alimentará a partir de la red eléctrica con una fuente de alimentación propia que se empaquetará, junto al resto de la unidad central, en una caja apropiada. Se dotará a la unidad central de radiofrecuencia de un interruptor de la alimentación y de un led indicador de su estado de funcionamiento (on/off).

A.5. Els nodes de vibradors

Los nodos de vibradores serán elementos independientes y autónomos que estarán aislados eléctricamente de cualquier otro elemento del sistema. Cada uno de ellos soportará uno o tres vibradores, según sea su tipo, y se alimentará de una batería propia capaz de soportar los picos de corriente necesarios para la operación de los vibradores así como de garantizar la autonomía de funcionamiento del nodo (pila de litio de 3 V tipo CR2). Asimismo, dispondrá de un interruptor que permita conectar y desconectar su alimentación.

Cada nodo de vibradores se basará en un microcontrolador nRF9E5 de Nordic, que servirá de puente entre el enlace de radiofrecuencia gestionado el transceptor de radiofrecuencia integrado en el microcontrolador y los vibradores del nodo (uno o tres, según sea su tipo), también gestionados a través de puertos de salida del microcontrolador.

Cada nodo de vibradores conocerá los identificadores de cinco bits de los distintos vibradores presentes en el nodo (uno o tres, según sea el tipo de nodo). Estos identificadores se modificarán al recibir una trama de indicación de los identificadores de los vibradores de los nodos activos del sistema vía el enlace de radiofrecuencia, a la cual se responderá inmediatamente con una trama de reconocimiento de la indicación de los identificadores de los vibradores de los nodos activos del sistema, tras el envío de la cual se pasará a un estado de nueva identificación completada. Durante este proceso de modificación de los identificadores de los vibradores de los nodos activos del sistema sólo deberá existir un nodo de vibradores en funcionamiento (alimentado).

El funcionamiento de los vibradores de un nodo estará gestionado por el microcontrolador a través de circuitos amplificadores apropiados. Cada vibrador de un nodo admitirá los siguientes cuatro estados de vibración:

- Vibrador en reposo (no vibración).
- Vibración de frecuencia 2 Hz y ciclo de trabajo del 20% (100 ms vibrando cada 500 ms).

- Vibración de frecuencia 5 Hz y ciclo de trabajo del 50% (100 ms vibrando cada 200 ms).
- Vibración de frecuencia 25 Hz y ciclo de trabajo del 50% (20 ms vibrando cada 40 ms).

El estado de vibración de los vibradores de un nodo se recibirá periódicamente vía el enlace de radiofrecuencia mediante las tramas de definición del estado de vibración de los treinta y un vibradores electromecánicos del sistema que la unidad central de radiofrecuencia transmite cada 100 ms.

El nodo de vibradores dispondrá, asimismo, de un LED tricolor (verde, rojo y anaranjado resultante de la combinación del verde y el rojo), que servirá para indicar el estado de funcionamiento del nodo (alimentación conectada o desconectada), el estado de los vibradores (algún vibrador del nodo vibrando o todos los vibradores del nodo en reposo), el estado de la pila (tensión de la pila aceptable o pila descargada).

Si la alimentación está desconectada, el LED estará permanentemente apagado. Si la alimentación está conectada y todos los vibradores del nodo están en reposo, el LED realizará una intermitencia de frecuencia 0,1 Hz y ciclo de trabajo del 1% (0,1 s encendido y 9,9 s apagado). Si la alimentación está conectada y alguno de los vibradores del nodo estuviese vibrando, el LED realizará una intermitencia de frecuencia 2 Hz y ciclo de trabajo del 20% (100 ms encendido y 400 ms apagado). Las intermitencias del LED serán de color verde si la tensión de la pila es aceptable y de color rojo si la pila está descargada (tensión inferior a 2,1 V).

El LED también servirá para indicar el estado de nueva identificación completada, estado en el que realizará una intermitencia de color anaranjado de frecuencia 0,1 Hz y ciclo de trabajo del 1% (0,1 s encendido y 9,9 s apagado). Además, en este estado se mantendrán todos los vibradores del nodo en reposo y se recomienda desconectar la alimentación del nodo lo antes posible a fin de no descargar la pila innecesariamente.

Para gestionar el funcionamiento de los vibradores, del LED y del transceptor de radiofrecuencia, el microcontrolador dispondrá de un reloj en tiempo real con un período de 10 ms. Todos los tiempos característicos del funcionamiento de los mencionados elementos deberán ser múltiplos de dicho período de tiempo.

A fin de disminuir el consumo del nodo, para así aumentar su autonomía de funcionamiento, tanto el microprocesador como el transceptor de radiofrecuencia integrados en el dispositivo nRF9E5 de Nordic estarán en modo de bajo consumo tanto tiempo como sea posible entre cada dos tramas consecutivas de definición del estado de

vibración de los treinta y un vibradores electromecánicos del sistema provenientes de la unidad central de radiofrecuencia.

Para ello, se sincronizará el reloj en tiempo real del microcontrolador con la recepción de las tramas de radiofrecuencia y, tras la recepción de una trama, tanto el microprocesador como el transceptor de radiofrecuencia estarán en modo de bajo consumo durante nueve períodos del reloj en tiempo real, estando el transceptor de radiofrecuencia en estado de recepción durante el décimo período de dicho reloj. En cualquier caso, se garantizará que el transceptor de radiofrecuencia entre en estado de recepción con la suficiente antelación respecto a la transmisión de una nueva trama (mínimo unos 0,5 ms) como para posibilitar su correcta recepción durante dicho período del reloj en tiempo real, siendo lo más habitual que se reciba en un lapso de tiempo comprendido entre los 3,5 y los 4 ms, tras los cuales tanto el microprocesador como el transceptor de radiofrecuencia podrán completar ese décimo período en estado de bajo consumo.

En caso de no recibirse ninguna trama durante el décimo período del reloj en tiempo real, el transceptor de radiofrecuencia entrará en estado continuo de recepción hasta la recepción de una trama. En esta situación, la recepción de una trama desembocará en una resincronización del reloj en tiempo real y se volverá al modo operativo normal.

Asimismo, se realizará una resincronización del reloj en tiempo real tras la recepción de un número predeterminado de tramas consecutivas (p.e., cada 256 tramas).

En los nodos de vibradores con un único vibrador, el vibrador se montará sobre el mismo circuito impreso sobre el que se montará su amplificador, el microcontrolador nRF9E5, una memoria E2PROM serie con el programa ejecutado por el microcontrolador, la antena del transceptor de radiofrecuencia, la batería, el interruptor y el LED. El conjunto se ubicará en una pequeña caja de plástico que presentará unas dimensiones aproximadas de $50 \times 37 \times 20$ mm.

En los nodos de vibradores con tres vibrador, cada uno de los vibradores se montará sobre un pequeño circuito impreso de unos 12×22 mm, que se encapsulará de forma apropiada para poder ser ubicado sobre la uña de uno de los dedos pulgar, índice y corazón de una mano. Las tres cápsulas satélite se conectarán mediante cables apropiados, de una longitud inferior a los 20 cm, al núcleo del nodo. Este núcleo contendrá, sobre una única placa de circuito impreso, los circuitos amplificadores de los tres vibradores, el microcontrolador nRF9E5, una memoria E2PROM serie con el programa ejecutado por el microcontrolador, la antena del transceptor de radiofrecuencia, la batería, el interruptor y el LED. El conjunto del núcleo se ubicará en una pequeña caja de plástico que presentará unas dimensiones aproximadas de $50 \times 37 \times 20$ mm.

La autonomía prevista para un nodo con un único vibrador será de unas 800 horas en estado de no vibración o bien de unas 14 horas en estado de vibración. La autonomía en funcionamiento normal dependerá del porcentaje de tiempo en el que el vibrador esté en estado de vibración. Para un porcentaje de tiempo en vibración del 10% del total, se estima una autonomía de unas 130 horas.

La autonomía prevista para un nodo con tres vibradores será de unas 800 horas en estado de no vibración o bien de unas 5 horas en estado de vibración (los tres vibradores vibrando continua y simultáneamente). La autonomía en funcionamiento normal dependerá del porcentaje de tiempo en el que los vibradores estén en estado de vibración. Para un porcentaje de tiempo en vibración del 10% del total para cada uno de los vibradores, se estima una autonomía de unas 45 horas.

A.6. La interfície en el PC

Se dispondrá de una interfaz en el PC que permita enviar a la unidad de radiofrecuencia los comandos que ésta interpreta y ejecuta (un carácter por el puerto serie, con el formato descrito en el apartado 4), esperar su reconocimiento (eco) y proceder a la retransmisión del comando en caso de no recibir el eco durante un tiempo apropiado.

B. Elecció de components dels mòduls actuadors

B.1. Elecció del mòdul de radiofreqüència

La banda ISM (*Industrial, Scientific and Medical*) acordada en les especificacions és la de 868 MHz. Aquesta banda és lliure, i és més que suficient per a la magnitud de les dades que haurem de transmetre. Tots els mòduls de radiofreqüència cercats han de permetre el treball en aquesta banda.

Un dels requisits més crítics que haurà d'acomplir el mòdul és el de mínim volum, per tant, l'encapsulat dels mòduls serà un factor decisiu per a la seva elecció.

A fi d'aconseguir minimitzar la mida del mòdul actuator, s'han buscat mòduls que integressin en un únic encapsulat un microcontrolador i un receptor o transceptor de RF. Aquesta varietat de productes és encara molt escassa en el mercat.

Un altre requisit que també ha resultat crític per a l'elecció del mòdul és el de mínim consum de corrent, ja que els actuadors han de ser autònoms i la duració de la bateria és indispensable per a la qualitat del sistema.

Respecte al rang de temperatures de funcionament del mòdul no hi ha requeriments específics, doncs aquest sistema funcionaria a temperatura ambient en l'interior de locals de climatologia mediterrània.

Finalment les característiques relatives a la sensibilitat, capacitat d'enviament d'informació, i el seu preu també han ajudat en l'elecció del mòdul de RF.

Partint d'aquestes valoracions s'ha realitzat un estudi de mercat a fi d'escollir l'element de RF més adequat a les necessitats del sistema. Les dades més significatives obtingudes en aquest estudi de mercat es mostren en les taules B.1.1. i B.1.2.

Partint d'aquesta comparativa, el mòdul que presenta millors característiques per a la incorporació en el sistema és el transceptor de RF fabricat per Nordic model nRF9E5, per la qual cosa ha estat escollit com a mòdul base del sistema.

Marca i model	Tipus	MCU	Encapsulat	Max. data rate [kbps]
Microchip [10] rfPIC12F675H	Transmitter	incorporat	20 pin SSOP 5,3x7,2mm	40
Atmel [1] AT86RF210	Transceiver	no incorporat	48 pin QFN 7x7mm	20
Chipcon [3] CC1010	Transceiver	incorporat	64 pin TQFP 10x10 mm	76,8
RFMonolithics Inc [15] DR5001	Receiver	no incorporat	17,8x17,8 mm	19,2
Radiometrix [14] Bim3	Transceiver	no incorporat	33x23x4 mm	64
Xemics [19] XE1202	Transceiver	no incorporat	44 pin LQFP 10x10mm	76,8
Pegasus [13] PTSS-2003 (versió europea)	Transceiver	incorporat	43,2x38,1x8,9 mm	76,8
NordIC [16] nRF903	Transceiver (config. sèrie)	no incorporat	32 pin TQFP 7x7mm	76,8
NordIC [18] nRF905	Transceiver (ShockBurst, reguladors de V)	no incorporat	32 pin QFN 5x5 mm	50
NordIC [18] nRF9E5	Transceiver (ShockBurst, reguladors de V)	incorporat	32 pin QFN 5x5 mm	50

Taula B.1.1. Resultats de l'estudi de mercat (característiques generals). Veure referències bibliogràfiques en la memòria

Marca i model	rang V [V]	Característiques de consum [mA]	Max. output power [dBm]
Microchip [10] rfPIC12F675H	2-5,5	a 3V, 434MHz, 6dBm: 4mA	+10
Atmel [1] AT86RF210	1,8-3,6	Recepció: 14,5mA Transmissió a 3,3V: 60mA	+12
Chipcon [3] CC1010	2,7-3,6	Recepció: 11,9mA Transmissió 4dBm: 23,5mA	+4
RFMonolithics Inc [15] DR5001	2,7-3,5	a 3V: 1,8mA	-
Radiometrix [14] Bim3	3-4	Recepció a 3V: 11,5mA Transmissió: 11mA	+3
Xemics [19] XE1202	2,4-3,6	Recepció: 14mA Transmissió: 48mA a 15dBm	+15
Pegasus [13] PTSS-2003 (versió europea)	2,9-5,5	Recepció: 30-60mA Transmissió: 400-500mA	+24
NordIC [16] nRF903	2,7-3,3	Recepció: 22,5mA Transmissió a -10dBm: 9mA	+10
NordIC [18] nRF905	1,9-3,6	Recepció: 12,5mA Transmissió a -10dBm: 11mA	+10
NordIC [18] nRF9E5	1,9-3,6	Recepció: 12,5mA Transmissió a -10dBm: 11mA	+10

Taula B.1.2. Resultats de l'estudi de mercat pel què respecta a les característiques elèctriques, amb $[dBm]=10 \times \log([mW])$ (Eq. B.1.1.)

Aquest mòdul, a més de les característiques descrites en el quadre (les quals han resultat completament determinants en la seva elecció davant les altres possibilitats, sobretot pel que respecta al seu volum), presenta altres característiques interessants:

- Ampli rang de temperatura de treball de -40 a $+85$ °C.
- Sensibilitat -100 dBm.
- Presenta un microcontrolador compatible amb l'estàndard 8051, amb RAM de 256 bytes, i memòria de programa de 4kbytes. La memòria de programa és de tipus RAM, per la qual cosa el programa ha de ser carregat des d'una EEPROM externa amb connexió sèrie. Aquesta peculiaritat és un inconvenient en quant a volum, però permet escriure dades registrades en temps d'execució de forma permanent (encara que es retiri la bateria). Moltes de les característiques del mòdul es poden controlar a través d'un conjunt de registres d'especial funcionalitat anomenats SFR.
- El *clock* de la CPU és generat a través d'un cristall, i pot ser configurat a 0.5MHz, 1MHz, 2MHz o 4MHz. També pot ser configurat a la freqüència del propi cristall.
- El transceptor inclou Shockburst, el qual afegeix automàticament preàmbul, adreça i CRC als missatges, i usa la codificació Manchester.
- El circuit incorpora també reguladors de voltatge, aconseguint més immunitat davant del soroll, i permet operar entre 1,9 i 3,6 V de subministrament.
- El nRF9E5 permet realitzar conversions A/D de 6, 8, 10 o 12 bits, doncs disposa d'un conversor A/D de 4 entrades a 80kbps que pot ser usat en mode diferencial. Aquest mòdul ofereix també la possibilitat de monitorització del voltatge que rep.
- Presenta diferents modes de baix consum. Entre ells, el mode de *power down* presenta un consum de $2,5\mu\text{A}$. I el seu consum en general és baix respecte als altres mòduls, tant en mode de recepció com en mode de transmissió.
- És compatible amb l'estàndard FCC CFR47 part '5 i amb ETSI EN 300 220-1 (normatives americana i europea respectivament).
- El seu preu és reduït, doncs costen uns 4 €/u (IVA no inclòs) al proveïdor Endrich Bauelemente.

B.2. Elecció del motor vibrador

El primer motor amb el qual es van fer proves va ser l'avisador per vibració Cebek C-6070.

Entre les característiques que presenta les primordials són el seu petit volum, i el seu voltatge de funcionament (1,3V).

Una característica crítica és el seu consum, doncs el motor és l'element de major consum del sistema (corrent de pic de 125mA, corrent mitjà de 75mA).

Una altra característica important és la seva velocitat mitjana, la qual és de 8000 rpm ± 1500 , així com la massa de la seva càrrega excèntrica. En aquest model, la càrrega presenta un radi de 2,8mm, i una longitud de $3 \pm 0,06$ mm.

I un altre factor menys important, però a tenir en consideració, és el grau de dissipació tèrmica que presenta el motor, ja que si se li imposa un cicle de treball gran, el motor s'escalfa amb rapidesa. Tot i així, segons el cicle de treball que s'usi això es podrà reduir i evitar problemes relacionats amb el sobreescalfament.

Finalment el preu és l'últim factor a comprovar. Aquest és a Onda Radio de 4,2€/u (sense incloure el 16% de IVA), força elevat en aquest cas.

Per experimentació (amb el muntatge descrit en l'apartat 0) s'ha determinat que la freqüència que produeix un major efecte sobre l'usuari oscil·la entorn els 5Hz. I un cicle de treball que podria ser un bon compromís entre consum i efecte sobre l'usuari es trobaria en el rang del 40-50%.

Finalment però per al sistema final s'ha escollit un vibrador diferent al de les primeres proves, en concret el corresponent al model 6SH-1708B de KOTL (firma representada per Endrich Bauelemente), el qual, presentant unes característiques semblants al model de Cebek, presenta certs trets diferenciadors que ens afavoreixen:

- Voltatge de funcionament de 2-3.8 V.
- Major radi (2,9 mm) i longitud de la càrrega excèntrica (4,2 mm), presentant una velocitat de rotació de 9000 rpm.
- El consum màxim de corrent és de 70 mA, molt menor al C-6070 de Cebek.
- Preu molt menor (de 0,5 €/u sense IVA)

B.3. Elecció de la bateria

Degut a la necessitat d'aconseguir el mínim volum ocupat possible es van buscar fonts d'alimentació que subministressin 3V, amb la màxima capacitat, i que fossin inicialment de botó.

A fi de poder estudiar el funcionament dels vibradors i les característiques de consum que presenten es van portar a terme unes primeres proves amb el circuit representat en les figures B.3.1 i B.3.2:

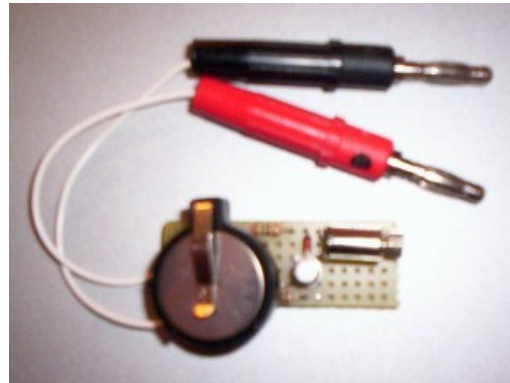
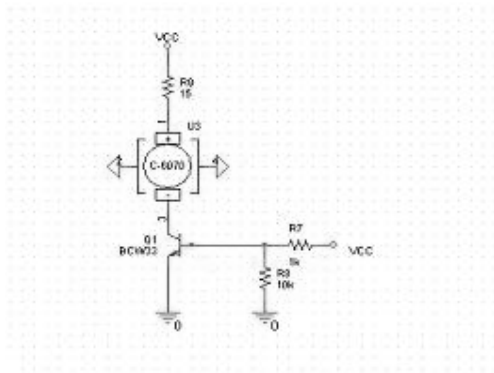


Fig. B.3.1. Representació del circuit en Orcad®

Fig. B.3.2. Fotografia del circuit

Els resultats que es van obtenir van ser els següents:

- Alimentant el mòdul mitjançant una pila de tipus botó (Varta 2032) de 3 V i capacitat 230mAh:
 - Es produeix una baixada molt brusca de tensió en l'arrancada del motor, i una pujada molt brusca en la seva parada. Un brusc increment de V en el règim transitori s'ha de corregir en el disseny final, ja que sinó es podria produir un reset del microcontrolador. La solució a aquest problema consisteix en afegir un filtre.
 - El motor en funcionament (en règim permanent) provoca una caiguda de voltatge de l'ordre de 0,7V.
 - El consum de corrent del vibrador és molt elevat. S'observa que en menys d'una hora de funcionament en règim continu la bateria utilitzada s'esgota. Encara que el funcionament del sistema mai seria en règim continu, sinó

amb un cert cicle de treball, es conclou que la capacitat d'aquesta bateria és insuficient si volem aconseguir una autonomia acceptable.

- Alimentant mitjançant una font de tensió, generant una ona quadrada entre 0 i 3V:
 - Es produeix una caiguda d'uns 0,4 V, dels quals uns 0,2 V són de caiguda instantània i la resta cau progressivament. El mòdul de radiofreqüència es podria veure afectat pel flanc de pujada de la tensió en la seva entrada, que es produeix en el moment de desconexió del motor.
 - La freqüència no influeix pràcticament en el consum, i s'ha determinat que de cara a l'usuari podria ser adequat un mode de funcionament a 5Hz, amb una vibració més sensible que a 10 Hz.
 - El cicle de treball és un factor determinant en el consum, i caldrà un compromís entre consum i potència de vibració. Un cicle de treball adequat oscil·laria entre $D=40-50\%$. En cas de necessitar un menor consum es podria pensar en disminuir-lo.
- Si s'incorpora un condensador de $22\mu\text{F}$ en paral·lel a la resistència de $10\text{k}\Omega$, es canvia la resistència de $10\text{k}\Omega$ per una de $100\text{k}\Omega$ i la resistència de $1\text{k}\Omega$ per una de $1,5\text{k}\Omega$:
 - Aquestes modificacions pretenen suavitzar l'increment de voltatge que es produeix en l'entrada del microcontrolador en el moment de desconexió del motor.
 - Mitjançant un oscil·loscopi digital es captura l'evolució del voltatge en funció del temps en una imatge. Així s'observa que amb aquesta nova modificació en aturar el motor el flanc de pujada del voltatge s'ha suavitzat, doncs ara es produeix un increment de $0,1\text{mV}$ en $1\mu\text{s}$.

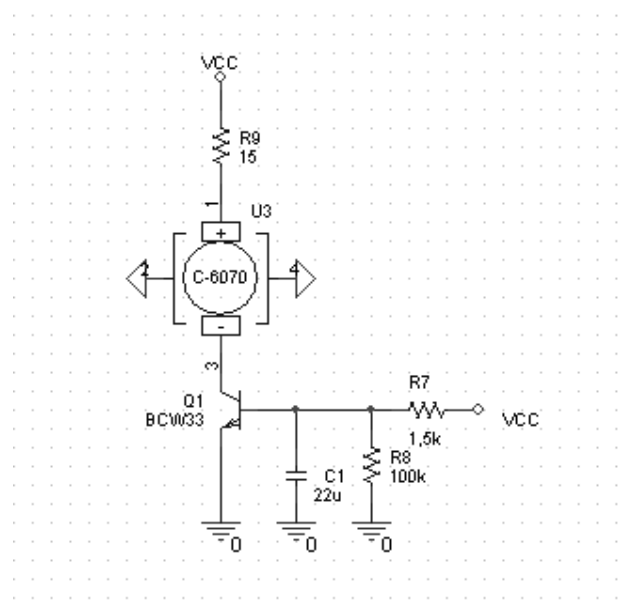


Fig. B.3.3. Representació del nou circuit en Orcad®

Per al disseny final, veient la problemàtica que presentaven les piles de botó degut a la seva baixa capacitat, s'ha escollit finalment una bateria del tipus CR2. Aquest tipus de bateries és de liti i s'usen especialment en fotografia, subministrant també 3V. Les seves dimensions poden variar lleugerament, però oscil·len entorn dels 15,6 mm de diàmetre, amb una alçada d'uns 27mm. El seu pes oscil·la entorn dels 11g i la seva capacitat nominal entorn els 850mAh.

Aquestes piles presenten un volum molt més elevat que les de botó, però permeten un millor compliment de les especificacions relatives a l'autonomia.

B.4. Elecció de l'EEPROM

Els criteris bàsics per a l'elecció de la memòria EEPROM són el seu encapsulat, que interessa amb mínim volum, la seva alimentació, preu i disponibilitat.

El tipus d'EEPROM ve marcat pel fabricant del mòdul de RF, el qual especifica que sigui del tipus 25320 amb interfície SPI.

L'EEPROM escollida finalment és del tipus 25AA320, ja que el rang de voltatge de funcionament és de 1,8V a 5,5V (l'alimentació ve determinada pel mòdul transceptor i és de 3V). Aquest tipus d'EEPROM treballa amb una freqüència de rellotge màxima de 1MHz i funciona correctament entre -40°C i $+85^{\circ}\text{C}$.

L'encapsulat escollit és el que ocupa menor superfície dels ofertats: TTSOP de 8 pins, que presenta unes mesures nominals de 3x6,38mm. El següent encapsulat en superfície és el tipus SOIC, de dimensions 4,9x6,02mm.

B.5. Elecció d'altres components

En aquest apartat es justificarà l'elecció de petits components de caràcter resistiu, capacitiu, LEDs, transistors i altres elements reguladors.

Molts d'aquests components vénen directament recomanats pel fabricant del mòdul de radiofreqüència en el cas que ens interessa: amb una antena de tipus espira, treballant a 868MHz.

S'ha escollit aquesta antena ja que permet una total integració en un receptacle, a diferència de les antenes unipolars, les quals suposarien la col·locació d'una antena exterior. Les antenes de tipus espira queden integrades en el propi circuit imprès. L'únic requeriment que imposen és que la zona ocupada per l'antena no s'usi com a base de cap altre component i ha de restar lliure.



R1	Resistència 1M Ω	1	Encapsulat SM-0603, 0.1 W, precisió +/- 1%
R2	Resistència 22k Ω	1	Encapsulat SM-0603, 0.1 W, precisió +/- 1%
R4	Resistència 10k Ω	1	Encapsulat SM-0603, 0.1 W, precisió +/- 1%
R5	Resistència 100k Ω	1	Encapsulat SM-0603, 0.1 W, precisió +/- 1%
R6	Resistència 18k Ω	1	Encapsulat SM-0603, 0.1 W, precisió +/- 1%
X1	Cristall de quars	1	16MHz, 42SMX, precisió +/-30ppm

Taula B.5.1. Components usats per al muntatge amb antena loop recomanats per Nordic

Disposició dels components:

- C1, C2 es col·loquen en els pins XC1 i XC2, i R1 en paral·lel al cristall de quars, a fi d'aconseguir més precisió en el senyal del rellotge.
- C3 i C4 en paral·lel es situen al pin VDD_PA com a desacobladors de voltatge, a fi que el senyal de l'antena no s'acobli a aquest senyal.
- C5, C6, C7, C8 actuen com a desacobladors del voltatge subministrat, a fi que el senyal de l'antena no s'acobli a la línia de l'alimentació. Aquest desacoblament ha de tenir lloc tan a prop com sigui possible dels pins VDD.
- S'ha prescindit de C9 i C10 recomanats per Nordic ja que actuen com a filtrat de l'entrada AREF (tensió de referència analògica externa).
- C11 estabilitza l'entrada de corrent a l'EEPROM 25AA320.
- C12, C13, C14 actuen com a sintonitzadors de l'antena a 868 MHz.
- A fi que el mòdul operi correctament, s'ha de col·locar una resistència de 22 k Ω (R2) entre el pin IREF i terra. Així es minimitza el biaix en el corrent de referència.
- R6 redueix el factor de qualitat de l'antena.

Altres components usats en el disseny (quantitats relatives als actuadors d'un únic vibrador), específics per a les nostres necessitats:



Referència	Component	Quantitat	Descripció
C15	Condensador 4.7 μ F	1	Tàntal 6V, encapsulat SM-1206
C16	Condensador 100nF	1	Tipus X7R ceràmic, tolerància +/-10%, encapsulat SM-0603
JP1, JP2	Connector de bateria	2	Tipus 108-2 retallat, ajustat a la mida de la capsa
Q1	Transistor BCW33	1	0.5A, 32V, 0.35W, $\beta_{min}=400$ Encapsulat SM-SOT23
R7	Resistència 3.9k Ω	1	Encapsulat SM-0603, 0.1 W, precisió +/- 1%
R8	Resistència 100k Ω	1	Encapsulat SM-0603, 0.1 W, precisió +/- 1%
U3	Motor avisador per vibració	1	KOTL 6SH-1708B
SW1	Interruptor miniaturitzat per a circuit imprès	1	SPDT, PC-TH T101MY9C/T101MH9C
D1	LED bicolor	1	3 mm, càtode comú, verd i vermell, de 3 pins, model L-93WEGW

Taula B.5.2. Altres components usats en el disseny (1 vibrador)

En el cas dels mòduls de tres vibradors, les quantitats utilitzades de cada component són les següents:

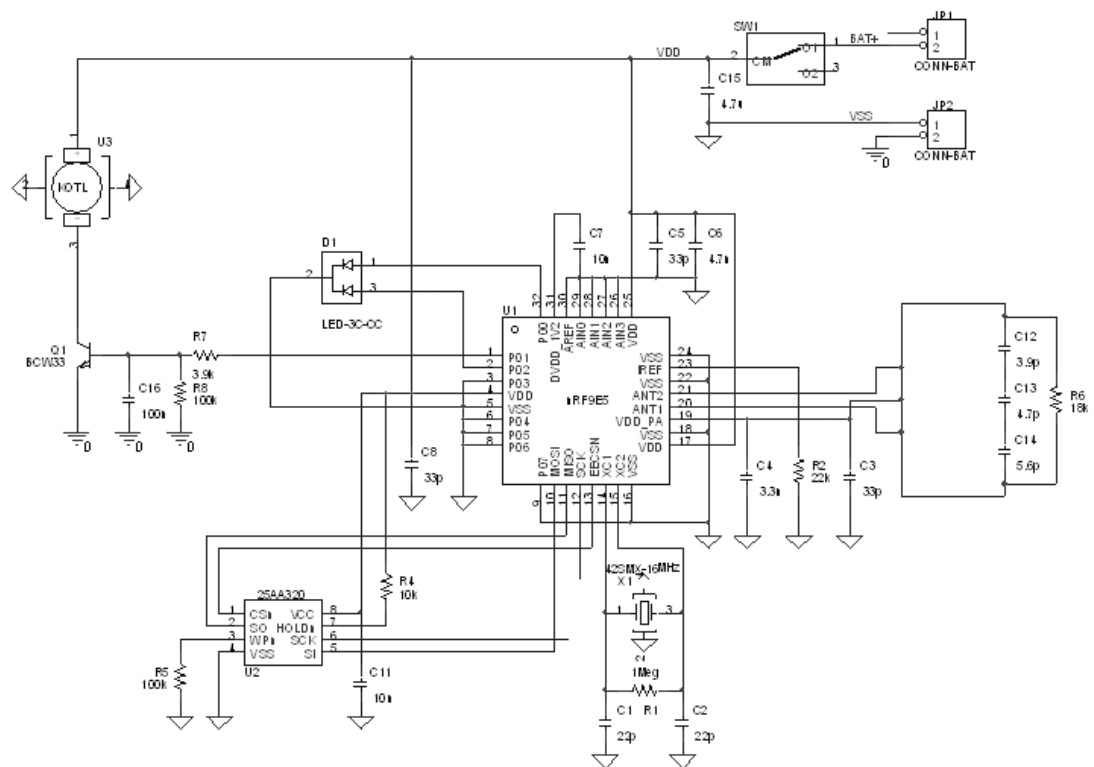
Referència	Component	Quantitat	Descripció
C15	Condensador 4.7 μ F	1	Tàntal 6V, encapsulat SM-1206
C16, C17, C18	Condensador 100nF	3	Tipus X7R ceràmic, tolerància +/-10%, encapsulat SM-0603
JP1, JP2	Connector de bateria	2	Tipus 108-2 retallat, ajustat a la mida de la capsa
Q1, Q2, Q3	Transistor BCW33	3	0.5A, 32V, 0.35W, $\beta_{min}=400$ Encapsulat SM-SOT23
R7,R9,R11	Resistència 3.9k Ω	3	Encapsulat SM-0603, 0.1 W, precisió +/- 1%
R8,R10,R12	Resistència 100k Ω	3	Encapsulat SM-0603, 0.1 W, precisió +/- 1%
U1,U2,U3	Motor avisador per vibració	3	KOTL 6SH-1708B
SW1	Interruptor miniaturitzat per a circuit imprès	1	SPDT, PC-TH T101MY9C/T101MH9C



D1	LED bicolor	1	3 mm, càtode comú, verd i vermell, de 3 pins, model L-93WEGW
----	-------------	---	--

Taula B.5.3. Altres components usats en el disseny (3 vibradors)

Així, els dissenys resultants són els següents:



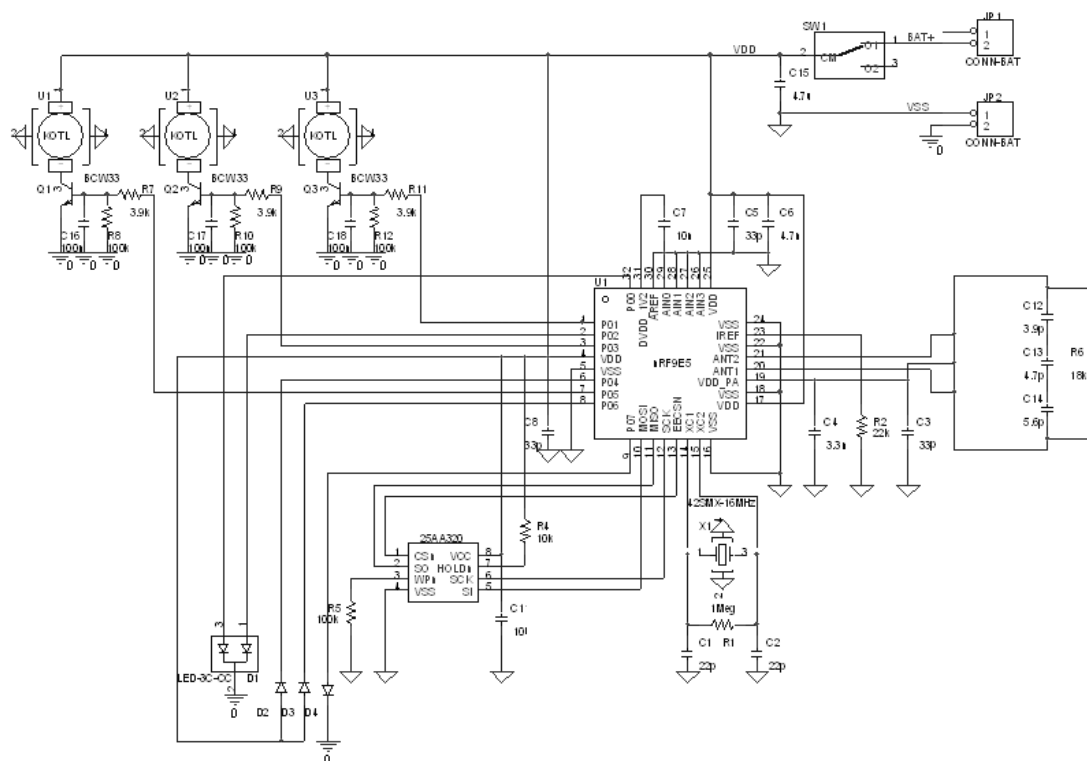


Fig. B.5.3. Disseny complet en Orcad® dels mòduls actuadors de 3 vibradors

Descripció dels components:

- JP1, JP2 són els connectors per a la bateria. Tenen forma de clip i serveixen també per a poder subjectar la pila correctament fins i tot en estat de vibració.
- SW1 és un microinterruptor que permet la connexió i desconnexió del mòdul.
- C15 actua com a filtre i permet estabilitzar el voltatge a l'entrada VDD del mòdul (pin 25), i també a l'entrada del motor vibrador. Aquesta estabilització és important, ja que evita el reset del mòdul davant de variacions brusques del voltatge a l'entrada del nRF9E5, especialment en el moment de desconnexió del motor.
- C16, R7 i R8 actuen com a filtre passa-baixos, de manera que quan a P01 emet un senyal graó el corrent a la porta del transistor Q1 pateixi variacions menys brusques, estabilitzant-lo. En el cas dels nodes actuadors de 3 vibradors aquest paper el fan respectivament C16, R7, R8 per a P01; C17, R9, R10 per a P03 i C18, R11, R12 per a P05.



- El transistor Q1 permet la connexió del motor vibrador davant d'una sortida 1 al pin corresponent a P00, actuant com a interruptor. A més evita també la necessitat d'un díode volant en paral·lel al motor.
- D1 és un led bicolor roig/verd. No cal cap resistència en sèrie ja que el corrent pels LEDs està limitat a 10mA pel buffer de sortida dels pins d'E/S del microcontrolador.

B.6. Elecció de l'encapsulat

El sistema es presenta recobert per una estructura de plàstic que té diferents finalitats:

- Suport i protecció del circuit imprès.
- Suport de la bateria.
- Protecció del motor. Si la càrrega excèntrica trobés un obstacle, aquest provocaria l'aturament del motor i es cremaria.
- Permetre una fàcil col·locació dels nodes sobre l'usuari.
- Millora la presentació del sistema.

En conseqüència s'ha buscat una caixa contenidora amb les mínimes dimensions possibles (50x37x20 mm) i amb recolzament per a circuit imprès.

Del muntatge de prova descrit anteriorment es va detectar un problema: les vibracions. Per a evitar que provoquessin males connexions amb la font d'alimentació s'han unit a la caixa uns clips per a subjectar amb força la bateria.





C. Elecció dels components de la unitat central

La unitat central té una certa quantitat d'elements comuns amb els mòduls actuadors, de manera que no es tornarà a justificar la seva elecció:

- L'EEPROM escollida és la mateixa que per al cas de les unitats actuadores
- El cristall de quars també coincideix, a 16MHz, 42SMX, i de precisió +/-30ppm
- El transceptor també és el mateix, ja que això és un requisit indispensable per a la comunicació entre els mòduls.
- L'antena en espira es configura de la mateixa manera: la freqüència de treball ha de ser la mateixa
- La majoria dels components coincideix, excepte els destinats a la zona dels ports P0. Els components en comú entre les unitats són els següents:

Referència	Component	Quantitat	Descripció
C1, C2	Condensador 22pF	2	Tipus NP0 ceràmic, tolerància +/-5%, encapsulat SM-0603
C3, C5, C8	Condensador 33pF	3	Tipus NP0 ceràmic, tolerància +/-5%, encapsulat SM-0603
C4	Condensador 3.3nF	1	Tipus X7R ceràmic, tolerància +/-10%, encapsulat SM-0603
C6	Condensador 4.7nF	1	Tipus X7R ceràmic, tolerància +/-10%, encapsulat SM-0603
C7, C11	Condensador 10nF	1	Tipus X7R ceràmic, tolerància +/-10%, encapsulat SM-0603
C12	Condensador 3.9pF	1	Tipus NP0 ceràmic, tolerància +/-0.1p, encapsulat SM-0603
C13	Condensador 4.7pF	1	Tipus NP0 ceràmic, tolerància +/-0.1p, encapsulat SM-0603
C14	Condensador 5.6pF	1	Tipus NP0 ceràmic, tolerància +/-0.1p, encapsulat SM-0603
R1	Resistència 1M Ω	1	Encapsulat SM-0603, 0.1 W, precisió +/- 1%
R2	Resistència 22k Ω	1	Encapsulat SM-0603, 0.1 W, precisió +/- 1%
R4	Resistència 10k Ω	1	Encapsulat SM-0603, 0.1 W, precisió +/- 1%
R5	Resistència 100k Ω	1	Encapsulat SM-0603, 0.1 W, precisió +/- 1%



C15	Condensador 4.7uF	1	Tàntal 6V, encapsulat SM-1206
-----	----------------------	---	----------------------------------

Taula C.1. Components comuns amb els nodes actuadors

En aquest cas també és necessari un connector a la font d'alimentació adequat i un interruptor qualsevol.

Una possible proposta per a la alimentació del mòdul és la utilització de regletes de connexió de 2 contactes i un alimentador de corrent estabilitzada a 3V i de corrent màxima subministrada per exemple de 100mA. Les proves s'han realitzat amb un alimentador a 3V i de corrent màxima de 350mA.

En el cas de la unitat central, l'espai no és un requeriment crític. És per això que no fa falta especificar el tipus d'encapsulat del component, hi ha flexibilitat. L'elecció del format es podrà realitzar segons la disponibilitat de components i/o segons el preu.

En la taula següent s'inclouen els components específics per a la unitat central que no són comuns amb els nodes actuadors, i una proposta d'encapsulat que pot ser utilitzat o no:

Referència	Component	Quantitat	Descripció
R7,R8	Resistència 100Ω	2	Encapsulat SM-0603, 0.1 W, precisió +/- 1%
R9	Resistència 10kΩ	1	Encapsulat SM-0603, 0.1 W, precisió +/- 1%
L1,L2	Inductància 15uH	2	Encapsulat SM-1812, precisió +/- 2%, corrent de saturació >350mA, R<1Ω
C16, C18	Condensador 1uF	2	Tàntal, tolerància +/- 20%, 20V, encapsulat SM-3216
C17	Condensador 0,47uF	1	Tàntal, tolerància +/- 20%, 35V, encapsulat SM-3216
RS232	Connector RS232	1	Connector femella per al port sèrie de 9 pins, DSUB
U3	Transceptor RS-232 Maxim	1	Transceptor MAX3218CAP, encapsulat SSO-20
C19	Condensador 100uF	1	Condensador electrolític, tolerància +/- 20%, 6,3V, encapsulat SMD
D4	Díode	1	BD6050, encapsulat SOT-23D
D1,D2,D3	LED	3	LED 3mm verd

Taula C.2. Altres components usats en el disseny



El LED D1 té com a finalitat actuar com a indicador de què el mòdul es troba alimentat. Els LEDs D2 i D3 serveixen com a indicadors del tipus de missatge que està enviant la unitat central. En cas que s'estigui enviant un missatge de comanda s'il·luminarà el LED D2, i en cas que el missatge sigui d'actualització d'estat s'il·luminarà el LED D3.

Finalment, l'element que actua com a enllaç entre el connector del port sèrie i el mòdul nRF9E5 és un circuit integrat MAXIM MAX3218CAP. Aquest permet una alimentació entre 1,8V i 4,25V i garanteix un *data rate* de 120kbps, de manera que és compatible amb softwares comuns i permet la comunicació amb el computador. Si el mòdul no percep un senyal d'entrada de nivell vàlid s'entra en mode *shut down* i el seu consum es redueix a 1µA. El sistema s'engega de nou quan es rep un nivell vàlid de senyal a l'entrada RS-232.

Cal recordar que en general es considera en RS232 un nivell lògic alt entre -3 i -15 V. Un nivell lògic baix es donarà entre $+3$ i -15 V. En general, els voltatges usats són $+12$ V i -12 V.

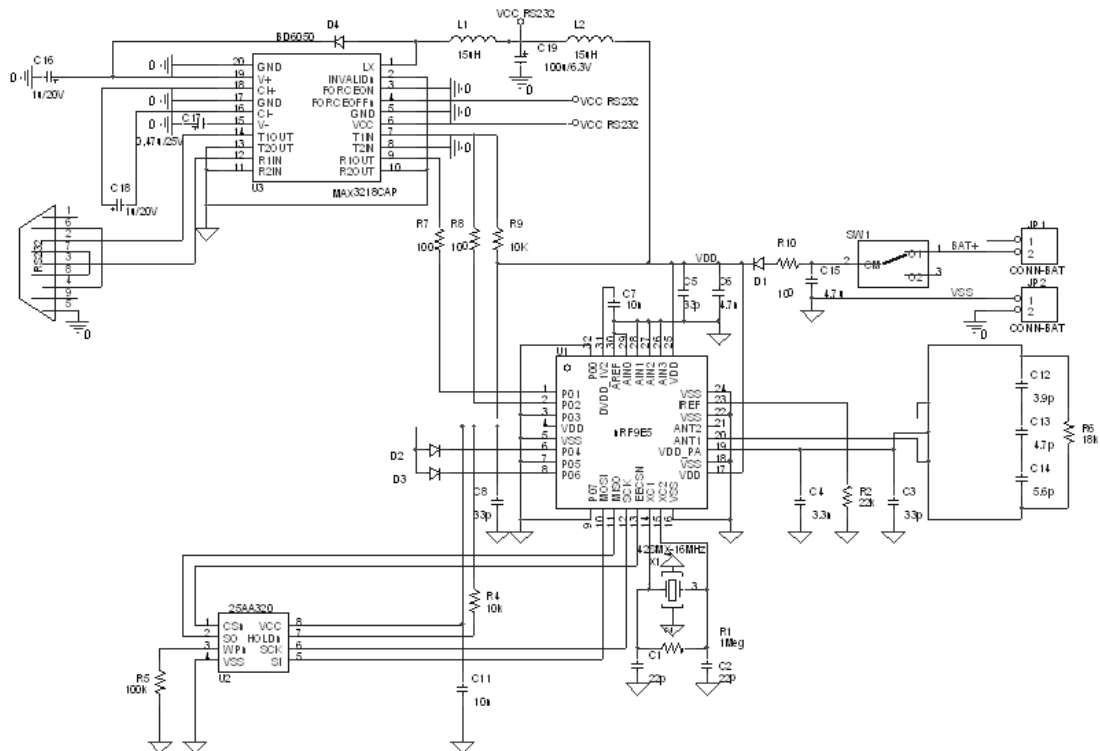


Fig. C.1. Disseny de la unitat central





D. Elecció dels components del mòdul de programació de l'EEPROM

Aquest hardware té com a finalitat permetre la programació de l'EEPROM escollida (25AA320).

Els components utilitzats per a l'elaboració d'aquest mòdul són els que s'indiquen en la taula següent:

Referència	Component	Quantitat	Descripció
C14	Condensador 10nF	1	Condensador ceràmic tipus X7R, tolerància +/-10%, 50V,encapsulat SM-0603
R3	Resistència 10k Ω	1	Encapsulat SM-0603, precisió +/- 1%
R4	Resistència 33k Ω	1	Encapsulat SM-0603, precisió +/- 1%
R5,R6,R10	Resistència 100k Ω	3	Encapsulat SM-0603, precisió +/- 1%
R7	Resistència no fitada	1	Encapsulat SM-0603
R8	Resistència 0 Ω	1	Encapsulat SM-0603, precisió +/- 1%
JP1	Connector	1	Connector femella de 14 pins
JP2	Connector per al programador	1	Connector femella de 10 pins
JP3	Connector	1	Connector femella de 2 pins
U1	EEPROM	1	4kbytes EEPROM amb interfície SPI, model 25AA320, encapsulat SO-8
U4A	Porta O	1	Porta O tipus 74LVC1G32 amb encapsulat SOT353-5
U9A	Buffer de bus	1	Buffer tipus 74LVC1G125 amb encapsulat SOT353-5
C15	Condensador 4.7 μ F	1	Tàntal 6V, encapsulat SM-1206

Taula D.1. Components del mòdul programador



El circuit resultant és el mostrat en la figura següent:

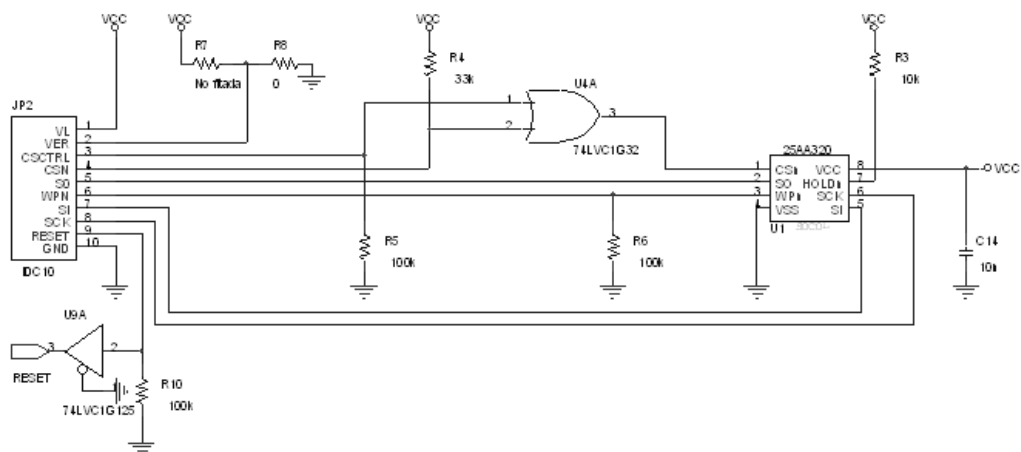


Fig. D.1. Disseny del mòdul programador



E. Impacte ambiental

Per a fer una avaluació de l'impacte ambiental del projecte s'analitzaran diferents aspectes a tenir en compte:

- Impacte sobre l'ambient de la fabricació dels circuits integrats i impresos.
- Emissions químiques que produeix la utilització dels components electrònics.
- Impacte ambiental dels elements d'alimentació.
- Impacte dels motors avisadors per vibració.
- Final de la vida dels components. Alternatives de reducció de l'impacte degut a aquesta fase.
- Emissió de radiacions electromagnètiques deguda a l'ús dels aparells de radiofreqüència.

E.1. Impacte de la fabricació dels circuits integrats i impresos

Per a la fabricació tant de circuits integrats com de circuits impresos hi intervenen molts elements metàl·lics, com per exemple alumini, coure, ferro, *kaptan* (fulla de poliamida), diferents tipus d'acers inoxidable i d'acers, *vespel*, entre d'altres. Tots ells han estat recoberts amb olis protectors durant la seva fabricació i manipulació a fi de proporcionar una protecció contra agents corrosius i per facilitar la seva manipulació i transport [García, M. J., *Medi ambient i tecnologia*, 1998 Edicions UPC].

En el moment d'utilització dels elements metàl·lics s'ha de realitzar un rentat previ a fi d'eliminar els olis protectors i poder aprofitar completament la seva conductivitat. Els mètodes que permeten efectuar aquest procés han evolucionat al llarg de la història. Històricament s'havia utilitzat un vapor desengreixant anomenat freó TMS (format per CFC-113, metil alcohol i additius). En un primer moment es va eliminar l'ús del CFC-113 i es va substituir per TCA (1-1-1 tricloroetà), però a l'any 1990 es va recomanar evitar l'ús tant del CFC-113 com del TCA, de manera que actualment es plantegen altres alternatives per a efectuar aquest rentat.

Les alternatives plantejades són terpens, barreges d'hidrocarburs, alcohol isopropil i sabons aquosos. Cadascun d'aquests elements alternatius té els seus avantatges i els seus



inconvenients. Uns d'ells són inflamables, altres són escassos i altres (seria el cas dels sabons aquosos) són recursos que es troben limitats per polítiques que es caracteritzen pel respecte al medi ambient.

En diverses proves s'ha demostrat que per a efectuar aquesta eliminació d'olis no és necessari recórrer als TCA, però les altres alternatives presenten problemes mediambientals greus.

Un altre element que s'ha d'eliminar del procés de fabricació de circuits impresos és el plom, el qual és un dels elements principals en el procés de soldatge de dispositius electrònics. S'han trobat algunes alternatives per al plom que presenten una menor toxicitat, com per exemple conductors adhesius, o tecnologies lliures de plom (*lead free*). La millor de les alternatives és la de la utilització de conductors adhesius, ja que el contingut en metalls preciosos d'aquest tipus de conductors fa que el seu reciclatge sigui més rendible que els metalls d'aliatges lliures de plom.

En estudis comparatius entre diferents tipus de conductors adhesius s'ha determinat que uns dels que presenten millor rendibilitat entre producció, reciclatge i qualitat són els de plata (*silver-epoxy*). Les alternatives trobades són més costoses però més respectuoses per a l'entorn.

E.2. Emissions químiques degudes a la utilització dels components electrònics

Els processos de fabricació de productes electrònics provoquen un impacte ambiental a l'entorn exterior, però també en provoquen un impacte sobre entorns tancats. Els productes electrònics presenten emissions químiques que poden provocar malalties com per exemple el síndrome de l'edifici malalt. Tot i així aquests efectes encara no són molt coneguts ja que els fabricants no han analitzat suficientment l'impacte dels seus productes en l'entorn en el qual s'utilitzaran.

En conseqüència és necessari que els fabricants realitzin una selecció de components que minimitzi l'impacte de la fabricació tant en els entorns oberts com en els tancats.

Per exemple, els ordinadors, impresores,... poden generar productes químics orgànics volàtils (COV com ozó, toluè, benzè, amoníac, acetona, etc.) i partícules nocives, que poden afectar la qualitat de l'aire respirable, amb conseqüències adverses per a la salut com irritació d'ulls i de vies respiratòries, càncers i al·lèrgies.



E.3. Impacte ambiental de les bateries dels nodes actuadors

Les piles utilitzades per als nodes són de tipus CR2, cilíndriques i de liti, en concret pertanyen a la sèrie de piles primàries de diòxid de manganès-liti.

Les piles són uns elements que constitueixen una greu amenaça en el moment en que passen a ser residus sòlids, ja que contenen metalls pesats que acaben arribant al medi. La pila que s'ha utilitzat, en concret conté liti i esdevé un risc quan és abocat o incinerat, moment en què s'assoleix el medi. Així, es contaminen els llacs, rius, aigües subterrànies i l'aire quan hi ha incineració.

Segons la directiva del parlament Europeu i del Consell en bateries i acumuladors, i en residus de bateries i acumuladors del 24/11/2003 SEC(2003) 1343, l'any 2002 el 45,5% de les bateries de tipus portable venudes als 15 de l'UE van ser tractades com a un residu final per incineració o abocament, enlloc de ser captades i reciclades.

El mercuri, plom i cadmi són de lluny les substàncies més problemàtiques com a residus. Aquests metalls no s'utilitzen en la bateria escollida, de manera que s'ha pogut aconseguir un producte menys perillós per al medi. Tot i així, en el cas del liti cal que les bateries no siguin tractades com a residus domèstics ordinaris.

La Directiva 2000/76/EC sobre incineració de residus estableix uns límits d'emissió cap a l'atmosfera d'elements contaminants que ha de comportar una disminució de les emissions de metalls pesats, ja que empitjoren la qualitat de l'aire.

Un 75% dels residus de bateries són enterrats en el sòl. Pel què respecta a l'abocament de les piles de liti hi ha una característica especial, que és que presenten risc d'explosió.

Per tant, caldrà que les piles que es gastin siguin portades als punts de recollida de piles més propers, a fi que puguin ser tractades convenientment.

En concret, si s'opta pel reciclatge, es presenten els següents beneficis per al medi:

- Menys bateries són incinerades o abocades.
- Es redueix la quantitat de metalls verges usats, de manera que es consumeix menys energia i ajuda a tancar el cicle de material.
- En les primeres etapes de la vida de les bateries es produeixen emissions per aire i contaminació per aigua. Quan els materials són reciclats aquestes emissions s'eviten.



Mitjançant el reciclatge s'aconsegueix capturar un gran ventall de les substàncies contingudes en les piles, de manera que enlloc de passar a ser uns residus municipals es passen a unes instal·lacions específiques per a les bateries usades.

L'ús de materials reciclats enlloc de metalls verges permet reduir l'energia primària utilitzada i la pol·lució generada per la mineria. Cal dir que això és molt important ja que la producció de metalls primaris és la font d'aproximadament el 10% de l'emissió global de CO₂.

Els impactes negatius sobre al medi que pot tenir el reciclatge són els referents al transport de les bateries recol·lectades i reciclades.

Els beneficis econòmics aconseguits gràcies al reciclatge de tots tipus de bateries són els següents:

- Reducció en el cost de l'energia en la producció de metalls verges.
- Reducció en el cost de la producció del metall verge.
- Estalvi en els costos de tractament dels residus finals, sobretot els deguts als abocaments.
- Si s'augmenta la quantitat de piles recollides es pot aconseguir reduir els costos del reciclatge segons les economies d'escala.
- S'eviten els costos de reduir el mal causat al medi per les bateries abocades.

Els costos generats per la recollida de piles i el seu reciclatge depenen en primer lloc del sistema de recollida. Per exemple, la separació magnètica seria una bona manera per a reduir els costos. Els costos totals es podran dividir en les següents categories:

- Variables:
 - Equipament dels punts de recollida.
 - Logística de la recollida.
 - Classificació.
 - Transport.
 - Reciclatge.



➤ Costos fixos:

- Administració.
- Relacions públiques i comunicació.

Els costos mitjos d'aquest reciclatge oscil·len entre els 400-900€/t.

Els costos de la incineració o abocament oscil·len entre els 50-120€/t.

Els costos del reciclatge depenen dels següents factors:

- Tipus de bateria.
- Tecnologia usada (hidrometal·lúrgia, pirometal·lúrgia, electrometal·lúrgia).
- De si la planta de reciclatge és o no dedicada.
- Del valor dels metalls recuperats. En concret, en el cas del liti, el ió Li és poc recuperable.

E.4. Final de la vida dels components

A fi de minimitzar l'impacte dels components electrònic sobre l'entorn s'ha treballat a fi d'aconseguir una disminució de la quantitat de residus generats. Al final de la vida útil d'aquests components existeixen diferents opcions: la manipulació del producte, la renovació, la remanufacturació, el reciclatge, la recuperació, etc.

El ràpid avanç de les tecnologies ha provocat un augment de les tecnologies de recuperació i reciclatge dels productes obsolets. Per treure partit d'aquests productes s'han buscat altres mercats per a la seva distribució renovant-los i actualitzant-los o bé escollint i venent parts útils dels equips electrònics.

Els productes electrònics que esdevenen deixalles representen l'1% de deixalles sòlides. Si aquests productes no són ben tractats poden representar una greu amenaça per al medi ambient degut al seu contingut en plom, cadmi, níquel i altres materials pesats.

El reciclatge a gran escala però és encara molt difícil d'aconseguir degut a la gran varietat de components dels sistemes electrònics.



Un cop donada per finalitzada la vida d'un component electrònic (quan ja no hi ha possibilitat del reaprofitament descrit en els paràgrafs anteriors), es poden diferenciar 5 fases en el processament, cadascuna de les quals porta associada una certa despesa. Les fases són les següents:

- Recollida i transport des de l'usuari final fins a la planta de desmuntatge i reciclatge.
- Desmuntatge. Aquest procés es fa principalment de manera manual i implica grans despeses.
- Processament mecànic. Consisteix en fragmentar i separar els components. És un processament més econòmic que el desmuntatge manual però és més difícil arribar a aconseguir materials purs.
- Processament (reciclatge). Aquesta fase presentarà beneficis en cas que el valor dels materials sigui elevat. Per dur a terme un reciclatge seran apropiats materials com el plàstic, el vidre i el coure, i els guanys que s'aconsegueixin es destinaran a compensar els costos en les altres fases del tractament final.
- Abocament. Els costos relacionats amb aquesta etapa depèn de la quantitat i del tipus de residus que s'han d'abocar.

Els resultats aconseguits es podrien millorar de la següent manera:

- Reciclant plàstics.
- Reciclant vidre.
- Utilitzant tècniques més avançades en la mecanització i la separació d'elements a fi de recuperar més materials valuosos.
- Fent un ecodisseny de nous productes en els quals el temps de desmuntatge es pugui disminuir, la qual cosa comporta una reducció en els costos.

En l'actualitat, però, els costos en el procés global de reciclatge són superiors als beneficis.



E.5. Impacte degut a la transmissió per radiofreqüència

La transmissió d'informació per radiofreqüència es basa en l'ús de les ones electromagnètiques per a transportar els senyals, la qual cosa té un impacte ambiental que afecta a la salut pública, ja que podria arribar a provocar efectes biològics nocius sobre les persones.

En ser la radiació electromagnètica de tipus no ionitzant els seus efectes biològics són únicament de tipus tèrmic. Per a aprofundir en aquesta afirmació es pot recórrer a [García, M. J., *Medi ambient i tecnologia*, 1998 Edicions UPC]. Alguns estudis indiquen que els efectes sobre el cos humà davant d'una exposició durant un llarg període de temps davant d'una radiació per radiofreqüència de gran potència poden ser: cremades, hipertèrmia, cataractes i esterilitat en el cas dels homes.

A fi de prevenir aquests efectes s'han definit uns límits màxims d'exposició, per sota dels quals no existeix perill pel cos humà. Es detallen a continuació.

- Taxa d'Absorció Específica (també anomenada SAR *Specific Absortion Rate*) [W/Kg].

És una mesura de l'energia electromagnètica que pot ser absorbida per unitat de temps (potència electromagnètica) per una massa biològica.

D'aquesta taxa es diferencien dos casos, ambdós superiors per a freqüències superiors a 10 MHz:

- Exposició laboral: 8 hores al dia, amb una TAE $<0,4$ W/kg.
- Exposició del públic: 24 hores al dia, contínuament, amb TAE $<0,08$ W/kg.

El problema d'aquest paràmetre és que és difícil de mesurar, per això s'han definit els valors llindar.

- Valors llindar TLV (*Threshold Limit Value*) [W/m²] mesuren densitat de potència per unitat de superfície.

En aquest cas també s'estableixen en funció de la freqüència límits segons si l'exposició és laboral o del públic.



Segons [García, M. J., *Medi ambient i tecnologia*, 1998 Edicions UPC], els valors de TLV amitjanats per a qualsevol període de 6 minuts a freqüències entre 400MHz i 2GHz són:

Freqüència	Ambient laboral		Ambient públic	
	[W/m ²]	[mW/cm ²]	[W/m ²]	[mW/cm ²]
0.4-2GHz	f/40	f/400	f/200	f/2000
869,6MHz	21,74	2,174	4,348	0,4348

Taula E.5.1. Valors líndar TLV

La potència màxima de transmissió del transceptor utilitzat és de +10dBm ([dBm]=10xlog([mW])), ja que el fabricant aconsegueix amb l'estàndard ETSI EN 300 220-1. Per tant, en W la potència de transmissió màxima és de 10mW, però se li ha d'aplicar l'eficiència de l'antena, de manera que la potència emesa és molt menor.

En concret, aplicant l'eficiència de l'antena trobada en l'apartat de l'annex referent al càlcul de l'abast, de l'equació G.2.13 $\eta=0.0053353$.

Així, la potència radiada valdrà 0.053353mW. L'emissió és isotròpica (igual en totes direccions).

Per tant, si la TAE laboral val 0,4 W/Kg, es pot observar que sigui quina sigui la massa corporal del treballador mai s'arribarà a assolir aquest valor, ni tampoc la TAE d'exposició al públic de 0,08W/Kg (ni en el cas de nou nats).

Si s'estudien els valors de TLV, per exemple a una distància de treball d'1cm respecte la unitat central, s'obté que:

$$\frac{P_{radiada}}{S} = \frac{P_{radiada}}{4 \cdot \pi \cdot R^2} = \frac{0.053353 \cdot 10^{-3}}{4 \cdot \pi \cdot 0.01^2} = 0.04245 \text{ W / m}^2 \quad (\text{Eq. E.5.1.})$$

Com es pot observar aquest valor és també ordres de magnitud inferior als valors màxims establerts tant per a ambient públic com per ambient laboral. Per tant, s'acompleixen sempre les normatives fins i tot a distàncies curtes (fins i tot en els pitjors casos).



F. Pressupost i termini

F.1. Pressupost

A fi de poder efectuar un pressupost, en primer lloc es determinarà el cost del material necessari per a l'elaboració d'un sistema. Cal dir que un sistema es pot constituir segons les preferències del client, pot ser format pel nombre i tipus de nodes que es desitgin. S'han efectuat els càlculs per al pitjor cas, on el sistema consta 15 mòduls amb un únic vibrador i 2 mòduls de 3 vibradors. En primer lloc cal distingir entre:

➤ Pressupost del hardware que es desenvoluparia per al client:

- Hardware de la unitat central.
- Hardware dels nodes actuadors d'un vibrador.
- Hardware dels nodes actuadors de 3 vibradors.

Per a cadascun d'aquest hardwares s'ha de comptabilitzar:

- Cost dels PCB.
- Cost dels components.
- Cost del muntatge.
- Cost de la verificació.

➤ Pressupost del disseny, desenvolupament, hardware d'avaluació i de recolzament, i verificació:

- Cost de disseny i desenvolupament.
- Cost del hardware d'avaluació (placa de proves i Evaluation Kit).
- Cost d'altres hardwares (placa de programació de l'EEPROM).

A continuació es realitzarà un estudi del pressupost dels components relatius a cadascun dels elements dissenyats per al sistema:



F.1.1. Pressupost de la unitat central

Referència	Component	Quantitat	Descripció	Marca/ Preu unitari	Distribuïdor/ Codi
C1, C2	Condensador 22pF	2	Tipus NP0 ceràmic, tolerància +/-5%, encapsulat SM-0603	Kemet 0,03€/u	RS Codi RS 264-4501
C3, C5, C8	Condensador 33pF	3	Tipus NP0 ceràmic, tolerància +/-5%, encapsulat SM-0603	Kemet 0,05€/u	RS Codi RS 298-9337
C4	Condensador 3.3nF	1	Tipus X7R ceràmic, tolerància +/-10%, encapsulat SM-0603	SMD 0,041€/u	Farnell Codi 753-609
C6	Condensador 4.7nF	1	Tipus X7R ceràmic, tolerància +/-10%, encapsulat SM-0603	Kemet 0,03€/u	RS Codi RS 264-4589
C7, C11	Condensador 10nF	2	Tipus X7R ceràmic, tolerància +/-10%, encapsulat SM-0603	Kemet 0,03€/u	RS Codi RS 264-4595
C12	Condensador 3.9pF	1	Tipus NP0 ceràmic, tolerància +/-0.1p, encapsulat SM-0603	SMD 0,048	Farnell Codi 721-918
C13	Condensador 4.7pF	1	Tipus NP0 ceràmic, tolerància +/-0.1p, encapsulat SM-0603	SMD 0,048	Farnell Codi 721-920
C14	Condensador 5.6pF	1	Tipus NP0 ceràmic, tolerància +/-0.1p, encapsulat SM-0603	SMD 0,048	Farnell Codi 721-931
R1	Resistència 1MΩ	1	Encapsulat 0603, Ni-Cr,0.1W,precisió +/- 0.1%	0,70 €/u	Farnell Codi 398-7498
R2	Resistència 22kΩ	1	Encapsulat 0603, 0.1 W, precisió +/- 0.1%	D1C 0,22€/u	RS Codi RS 440-7506
R4	Resistència 10kΩ	1	Encapsulat 0603, 0.1 W, precisió +/- 0.1%	D1C 0,22€/u	RS Codi RS 440-7499
R5	Resistència 100kΩ	1	Encapsulat 0603, 0.1 W, precisió +/- 0.1%	D1C 0,22€/u	RS Codi RS 440-7534
R6	Resistència 18.2 o 17.8kΩ	1	Encapsulat 0603,Ni- Cr 0.1 W, precisió +/- 0.1%	0,67€/u	Farnell Codi 398-9677 (18.2) Codi 398-9665 (17.8)
C15	Condensador 4.7uF	1	Tàntal 16V, SMT,10V,10%	Kyocera 0,55€/u	RS Codi RS 464-9126
R7,R8,R10	Resistència 100Ω	3	Encapsulat 0603, 0.1 W, precisió +/- 0.1%	D1C 0,22€/u	RS Codi RS 440-7427



R9	Resistència 10kΩ	1	Encapsulat 0603, 0.1 W, precisió +/- 0.1%	D1C 0,22€/u	RS Codi RS 440-7499
L1,L2	Inductància 15uH	2	Sèrie PA, 10%, d'alta corrent per a muntatge superficial	PANASONIC 0,42€/u	RS Codi RS 180-7896
C16, C18	Condensador 1uF	2	Tàntal, tolerància +/- 10%, 20V, SMT	Kyocera 0,55€/u	RS Codi 464-9053
C17	Condensador 0,47uF	1	Tàntal, muntatge superficial T491 35V	Kemet 0,16€/u	RS Codi 262-4692
RS232	Connector RS232	1	Connector mascle per al port sèrie de 9 pins, DSUB	0,15€/u	Diotronic Codi DN9PZ
-	Cable de connexió	1	Cable de connexió per a port sèrie no creuat, 2metres	5,41€/u	Diotronic Codi CAB230
VDD	Mòdul alimentador	1	Alimentador 3V 350mA	TRQ 10,4€/u	Onda Radio Codi TCS3V
X1	Cristall de quars	1	16MHz, 42SMX, precisió +/-30ppm	0,83 €/u	Endrich Bauelemente
JP1,JP2	Regleta de connexió	1	Regleta de connexió de pas 5mm	0,14 €/u	Onda Radio Codi RBV2
SW1	Commutador	1	Commutador 1 circuit 3 posicions	0,4264	Diotronic Codi C1C3P
U1	Mòdul transceptor de RF	1	Model nRF9E5, a 868MHz Encapsulat SM- QFN-32	Nordic 4 €/u	Endrich Bauelemente
U2	Memòria EEPROM	1	25AA320 Encapsulat SM- SOIC-8 25320	XCR 7,45	Farnell Codi 788-685
U3	Transceptor RS-232 Maxim	1	Transceptor MAX3218CAP, encapsulat SSO-20	Maxim 2,4€/u	Maxim MAX3218CAP
RS232	Connector	1	Connectors SUB-D femella (9 pins) per a port sèrie	0,2651€/u	Diotronic Codi DN9SZ
C19	Condensador 100uF	1	Condensador electrolític, 6.3V, SMT muntatge superficial	Panasonic 3,51€/u	RS Codi RS 414-9474
D4	LED	1	LED BD6050, encapsulat SOT- 23D,200mA,0.7V	InOne 0,0242€/u	Farnell Codi 2332700
D1,D2,D3	LED	3	LED verd 3mm	SLI 0,48€/u	Farnell Codi 595-329

Taula F.1.1.1. Components per a la unitat central (preus sense IVA)



El pressupost total corresponent a una unitat central és de 42,4907€ (preu sense IVA).

F.1.2. Pressupost de les unitats actuadores d'un vibrador

Referència	Component	Quantitat	Descripció	Marca/ Preu unitari	Distribuïdor/ Codi
C1, C2	Condensador 22pF	2	Ceràmic multicapa, format SMT-0603, dielèctric COG, 50V, tolerància +/-5%,	Kemet 0,03€/u	RS Codi RS 264-4501
C3, C5, C8	Condensador 33pF	3	Ceràmic multicapa, format SMT-0603, dielèctric COG, 50V, tolerància +/-5%,	Kemet 0,05€/u	RS Codi RS 298-9337
C4	Condensador 3.3nF	1	Tipus X7R ceràmic, tolerància +/-10%, encapsulat 0603	SMD 0,041€/u	Farnell Codi 753-609
C6	Condensador 4.7nF	1	Ceràmic, format SMT-0603, dielèctric X7R, 50V, tolerància +/-10%,	Kemet 0,03€/u	RS Codi RS 264-4589
C7, C11	Condensador 10nF	1	Ceràmic, format SMT-0603, dielèctric X7R, 50V, tolerància +/-10%,	Kemet 0,03€/u	RS Codi RS 264-4595
C12	Condensador 3.9pF	1	NPO, ceràmic multicapa, tolerància +/-0.25pF, encapsulat 0603	SMD 0,048	Farnell Codi 721-918
C13	Condensador 4.7pF	1	NPO, ceràmic multicapa, tolerància +/-0.25pF, encapsulat 0603	SMD 0,048	Farnell Codi 721-920
C14	Condensador 5.6pF	1	NPO, ceràmic multicapa, tolerància +/-0.5pF, encapsulat 0603	SMD 0,048	Farnell Codi 721-931
R1	Resistència 1MΩ	1	Encapsulat 0603, Ni-Cr, 0.1W, precisió +/- 0.1%	0,70 €/u	Farnell Codi 398-7498
R2	Resistència 22kΩ	1	Encapsulat 0603, 0.1 W, precisió +/- 0.1%	D1C 0,22€/u	RS Codi RS 440-7506
R4	Resistència 10kΩ	1	Encapsulat 0603, 0.1 W, precisió +/- 0.1%	D1C 0,22€/u	RS Codi RS 440-7499
R5	Resistència 100kΩ	1	Encapsulat 0603, 0.1 W, precisió +/- 0.1%	D1C 0,22€/u	RS Codi RS 440-7534
R6	Resistència 18.2 o 17.8kΩ	1	Encapsulat 0603, Ni- Cr 0.1 W, precisió +/- 0.1%	0,67€/u	Farnell Codi 398-9677 (18.2) Codi 398-9665 (17.8)



X1	Cristall de quars	1	16MHz, 42SMX, precisió +/-30ppm	0,83 €/u	Endrich Bauelemente
C15	Condensador 4.7uF	1	Tàntal 16V, SMT, 10V, 10%	Kyocera 0,55€/u	RS Codi RS 464-9126
C16	Condensador 100nF	1	Ceràmic, format SMT-0603, dielèctric X7R, 50V, tolerància +/-10%,	Kemet 0,05€/u	RS Codi RS 264-4630
JP1, JP2	Connector de bateria	2	Tipus 108-2 retallat, ajustat a la mida de la capsa	0,1€/u	Keystone Catàleg 108-2
VDD	Pila	1	Tipus CR2, 3 V, 950mAh, Li-Mg, no recarregables	Duracell 7,68 €/u	Farnell Codi 207-550
Q1	Transistor BCW33	1	0.5A, 32V, 0.35W, $\beta_{min}=400$ Encapsulat SM-SOT23	SMD 0,152€/u	Fairchild Codi 934-290
R7	Resistència 3.92kΩ	1	Encapsulat 0603, Ni-Cr 0.1 W, precisió +/- 0.1%	0,67€/u	Farnell Codi 398-9033
R8	Resistència 100kΩ	1	Encapsulat SM-0603, 0.1W, precisió +/- 0.1%	D1C 0,22€/u	RS Codi RS 440-7534
U3	Motor avisador per vibració	1	Model 6SH-1708B a 3V	KOTL 0,5€/u	Endrich Bauelemente
U1	Mòdul transceptor de RF	1	Model nRF9E5, a 868MHz Encapsulat SM-QFN-32	Nordic 4 €/u	Endrich Bauelemente
U2	Memòria EEPROM	1	25AA320 Encapsulat SM-SOIC-8 25320	XCR 7,45	Farnell Codi 788-685
SW1	Interruptor miniaturitzat per a circuit imprès	1	SPDT, PCB Mounting T101MH9CBE, on-on	ITT 3,51€/u	Farnell Codi 986-010
D1	LED bicolor	1	3 mm, càtode comú, verd i vermell, de 3 pins, model L-93WEGW	Kingbright 0,67€/u	Farnell Codi 637-210
-	Capsa	1	Capsa de plàstic dur 50x37x20 mm	2 €/u	Onda Radio

Taula F.1.2.1. Components per als nodes actuadors d'un vibrador (preus sense IVA)

El pressupost total corresponent a un mòdul d'un vibrador és de 30,967€ (preu sense IVA).

F.1.3. Pressupost de les unitats actuadores de tres vibradors

Referència	Component	Quantitat	Descripció	Marca/ Preu unitari	Distribuïdor/ Codi
C1, C2	Condensador 22pF	2	Ceràmic multicapa, format SMT- 0603,dialèctric COG, 50V,tolerància +/-5%,	Kemet 0,03€/u	RS Codi RS 264-4501
C3, C5, C8	Condensador 33pF	3	Ceràmic multicapa, format SMT-0603, dialèctric COG, 50V, tolerància +/-5%,	Kemet 0,05€/u	RS Codi RS 298-9337
C4	Condensador 3.3nF	1	Tipus X7R ceràmic, tolerància +/-10%, encapsulat 0603	SMD 0,041€/u	Farnell Codi 753-609
C6	Condensador 4.7nF	1	Ceràmic, format SMT-0603, dialèctric X7R, 50V, tolerància +/-10%,	Kemet 0,03€/u	RS Codi RS 264-4589
C7, C11	Condensador 10nF	1	Ceràmic, format SMT-0603, dialèctric X7R, 50V, tolerància +/-10%,	Kemet 0,03€/u	RS Codi RS 264-4595
C12	Condensador 3.9pF	1	NP0, ceràmic multicapa, tolerància +/-0.25pF, encapsulat 0603	SMD 0,048	Farnell Codi 721-918
C13	Condensador 4.7pF	1	NP0, ceràmic multicapa, tolerància +/-0.25pF, encapsulat 0603	SMD 0,048	Farnell Codi 721-920
C14	Condensador 5.6pF	1	NP0, ceràmic multicapa, tolerància +/-0.5pF, encapsulat 0603	SMD 0,048	Farnell Codi 721-931
R1	Resistència 1MΩ	1	Encapsulat 0603, Ni-Cr,0.1W,precisió +/- 0.1%	0,70 €/u	Farnell Codi 398-7498
R2	Resistència 22kΩ	1	Encapsulat 0603, 0.1 W, precisió +/- 0.1%	D1C 0,22€/u	RS Codi RS 440-7506
R4	Resistència 10kΩ	1	Encapsulat 0603, 0.1 W, precisió +/- 0.1%	D1C 0,22€/u	RS Codi RS 440-7499
R5	Resistència 100kΩ	1	Encapsulat 0603, 0.1 W, precisió +/- 0.1%	D1C 0,22€/u	RS Codi RS 440-7534
R6	Resistència 18.2 o 17.8kΩ	1	Encapsulat 0603,Ni- Cr 0.1 W, precisió +/- 0.1%	0,67€/u	Farnell Codi 398-9677 (18.2) Codi 398-9665 (17.8)
X1	Cristall de quars	1	16MHz, 42SMX, precisió +/-30ppm	0,83 €/u	Endrich Bauelemente



C15	Condensador 4.7uF	1	Tàntal 16V, SMT,10V,10%	Kyocera 0,55€/u	RS Codi RS 464-9126
C16, C17, C18	Condensador 100nF	3	Ceràmic, format SMT-0603, dielèctric X7R, 50V, tolerància +/-10%,	Kemet 0,05€/u	RS Codi RS 264-4630
JP1, JP2	Connector de bateria	2	Tipus 108-2 retallat, ajustat a la mida de la capsa	0,1€/u	Keystone Catàleg 108-2
VDD	Pila	1	Tipus CR2, 3 V, 950mAh, Li-Mg, no recargables	Duracell 7,68 €/u	Farnell Codi 207-550
Q1, Q2, Q3	Transistor BCW33	3	0.5A, 32V, 0.35W, $\beta_{min}=400$ Encapsulat SM- SOT23	SMD 0,152€/u	Fairchild Codi 934-290
R7, R9, R11	Resistència 3.92k Ω	3	Encapsulat 0603, Ni- Cr 0.1 W, precisió +/- 0.1%	0,67€/u	Farnell Codi 398-9033
R8, R10, R12	Resistència 100k Ω	3	Encapsulat SM- 0603, 0.1W, precisió +/- 0.1%	D1C 0,22€/u	RS Codi RS 440-7534
U1, U2, U3	Motor avisador per vibració	3	Model 6SH-1708B a 3V	KOTL 0,5€/u	Endrich Bauelemente
U1	Mòdul transceptor de RF	1	Model nRF9E5, a 868MHz Encapsulat SM- QFN-32	Nordic 4 €/u	Endrich Bauelemente
U2	Memòria EEPROM	1	25AA320 Encapsulat SM- SOIC-8 25320	XCR 7,45	Farnell Codi 788-685
SW1	Interruptor miniaturitzat per a circuit imprès	1	SPDT, PCB Mounting T101MH9CBE, on- on	ITT 3,51€/u	Farnell Codi 986-010
D1	LED bicolor	1	3 mm, càtode comú, verd i vermell, de 3 pins, model L-93WEGW	Kingbright 0,67€/u	Farnell Codi 637-210
-	Suport de metacrilat	3	Suport semiesfèric. Veure fotografies	5 €/u	Laboratori comú d'enginyeria mecànica
-	Capsa	1	Capsa de plàstic dur 50x37x20 mm	2 €/u	Onda Radio

Taula F.1.3.1. Components per als nodes actuadors de tres vibradors (preus sense IVA)

El pressupost total corresponent a un mòdul de 3 vibradors és 49,151€ (preu sense IVA).

F.1.4. Pressupost del mòdul programador de l'EEPROM

Referència	Component	Quantitat	Descripció	Marca/ Preu unitari	Distribuïdor/ Codi
C14	Condensador 10nF	1	Ceràmic, format SMT-0603, dielèctric X7R, 50V, tolerància +/-10%,	Kemet 0,03€/u	RS Codi RS 264-4595
R3	Resistència 10kΩ	1	Encapsulat SM- 0603, precisió +/- 1%	D1C 0,22€/u	RS Codi RS 440-7499
R4	Resistència 33kΩ	1	Encapsulat SM- 0603, precisió +/- 1%, 0.1W	TYCO 0,05€/u	RS Codi RS 213-2474
R5,R6, R10	Resistència 100kΩ	2	Encapsulat SM- 0603, precisió +/- 1%	D1C 0,22€/u	RS Codi RS 440-7534
R7	Resistència no fitada (per exemple de 100Ω)	1	Encapsulat 0603, 0.1 W, precisió +/- 0.1%	D1C 0,22€/u	RS Codi RS 440-7427
R8	Resistència 0Ω	1	Encapsulat SM- 0603, precisió +/-1%, 0.1W	Tyco 0,05€/u	RS Codi RS 213-1982
JP1	Connector	1	Connector femella de 14 pins	0,72€/u	Diotronic Codi 636.1441
JP2	Connector per al programador	1	Connector femella recte perfil doble fila de 10 pins	0,132€/u	Onda Radio Codi CO 1610
-	Connector	1	Connector 2 contactes mascle Jumper de pas 2,54mm i longitud 6mm	0,0152€/u	Onda Radio Codi CO 3008 N
JP3	Connector	1	Connector femella de 2 pins de pas 2,54mm i 2 files	0,319€/ tira de 80 pins	Onda Radio Codi CO 3080
U4A	Porta O	1	Porta O tipus 74LVC1G32 amb encapsulat SOP-5	Texas Instruments 0,382€/u	Farnell Codi 7453516
U1	Sòcol	1	Sòcol de pressió d'inserció nul·la per SOIC-8	3M Group 25,38€/u	RS Codi RS 766-980



U9A	Buffer de bus	1	Buffer tipus 74LVC1G125 amb encapsulat SOT23-5	Texas instruments 0,0955€/u	Farnell Codi 4830027
-----	---------------	---	--	--------------------------------	-------------------------

Taula F.1.4.1. Components per al mòdul programador (preus sense IVA)

El pressupost total corresponent a un mòdul programador d'EEPROM és de 28,2737€ (preu sense IVA).

F.1.5. Pressupost del hardware de desenvolupament i de proves

La principal partida d'aquest pressupost és el corresponent a l'*Evaluation Kit* de NordIC, kit per a desenvolupament d'aplicacions amb el mòdul nRF9E5, el qual té un cost de 457€ (sense IVA).

S'ha adquirit també un alimentador estabilitzat de 3V amb corrent màxima subministrada de 350mA. Aquest ha tingut un cost de 10,40€ (sense IVA), i un cable connector per a port sèrie no creuat de 2 metres de 5,41€ (sense IVA).

Tal i com s'ha explicat en la memòria s'ha elaborat una placa de proves. El detall dels components que s'han escollit i el corresponent preu es mostra en la taula següent:

Referència	Component	Quantitat	Descripció	Marca/ Preu unitari	Distribuïdor/ Codi
C1, C2, C3	Condensador 100nF	3	Condensador multicapa 100nF 63V 10%	0,0293€/u	Onda Radio Codi TH100K63
C4	Condensador 4,7uF	1	Condensador electrolític 63V 4,7uF	0,0215€/u	Onda Radio Codi JR0634U7
R1, R2, R3	Resistència 3,9kΩ	3	Resistència 0,25W, 5%	0,0052€/u	Onda Radio Codi PR253K9
R4, R7, R10	Resistència 100kΩ	3	Resistència 0,25W, 5%	0,0052€/u	Onda Radio Codi PR25100K
R5, R8, R11	Resistència 13Ω	3	Resistència 0,25W, 5%	0,0052€/u	Onda Radio Codi PR2513H
R6, R9, R12	Resistència 100Ω	3	Resistència 0,25W, 5%	0,02€/u	Radio Watt



Q1, Q2, Q3	Transistor BCW33	3	0.5A, 32V, 0.35W, $\beta_{\min}=400$ Encapsulat SM-SOT23	SMD 0,152€/u	Fairchild Codi 934-290
U1, U2, U3	Motor avisador per vibració	3	Micro-motor Cebek C-6070 1,3V, 11,7 Ω	Cebek 3,78€/u	Onda Radio Codi C6070
D1, D2, D3	LEDs	3	LED 5mm incolor (vermell)	0,16€/u	Radio Watt
D4	LED	1	LED 5mm groc	0,16€/u	Radio Watt
D5	LED	1	LED bicolor 5mm càtode comú TLUV5300	0,2€/u	Diotronic Codi CQX95
D6	LED	1	LED 5mm vermell	0,16€/u	Radio Watt
D7	LED	1	LED 5mm alta lluminositat verd	0,168€/u	Diotronic Codi HLMP3519
SW1	Commutador	1	Commutador 3 posicions 1 circuit	0,4264€/u	Diotronic Codi C1C3P
SW2	Commutador	1	Commutador 2 posicions 1 circuit	0,1€/u	Radio Watt
JP1, JP2	Regleta de connexió	1	Regleta de connexió de pas 5mm	0,14 €/u	Onda Radio Codi RBV2
JP3	Connector	1	Connector femella de 14 pins	0,72€/u	Diotronic Codi 636.1441
VDD	Mòdul alimentador	1	Alimentador 3V 350mA	TRQ 10,4€/u	Onda Radio Codi TCS3V
-	Cable de connexió	1	Cable de connexió per a port sèrie no creuat, 2metres	5,41€/u	Diotronic Codi CAB230
-	Cable	1	Cable de connexió 0,25 marró	0,368€/u	Onda Radio CC1252M
-	Cable	1	Cable de connexió 0,25 vermell	0,4€/u	Radio Watt
-	Placa de topes	1	Placa baquelita CT3	3,8€/u	Radio Watt
-	Cable pla	1	Cable pla gris 14C ASSMAN C AWG-28	0,38€/u	Onda Radio Codi AWG2814
-	Connector	2	Connectors mascle recte sota perfil doble F 14 pins	0,2333€/u	Onda Radio Codi CO1614
-	Connector	0	Connector de 3 femelles de 2 pins de pas 2,54mm i 2 files. Veure unitat central.	0,319€/ tira de 80 pins (està comptabilitzada ja en la unitat central)	Onda Radio Codi CO 3080
-	Connector	3	Connector 2 contactes mascle Jumper de pas 2,54mm i longitud 6mm	0,0152	Onda Radio Codi CO 3008 N

Taula F.1.5.1. Components per al mòdul de proves elaborat (preus sense IVA)



El pressupost total corresponent al mòdul de procés és de 35,8368€ (preu sense IVA).

Pel què respecta al kit d'avaluació, s'han pagat 457€ (sense IVA).

A aquests costos cal afegir els costos d'altres materials que s'han anat adquirint per a realitzar proves que no han format part de la placa de proves. El cost sense IVA d'aquesta partida ha estat de 29,5€.

Així, el cost total de les proves ha estat de 522,3368 € (sense IVA).

F.1.6 Càlcul del cost total

En primer lloc es realitzarà el càlcul del cost total sense IVA del hardware del projecte:

➤ Cost total dels components:

- Unitat central: 42,4907€ (1 per sistema).
- Mòdul d'1 vibrador: 30,967€ (15 per sistema).
- Mòdul de 3 vibradors: 49,151€ (2 per sistema).
- Mòdul de programació de l'EEPROM: 49,151€ (1 per a tots els sistemes).
- Placa de proves: 35,8368€ (1 per a tots els sistemes).
- Cost d'altres elements per a proves: 29,5€ (1 per a tots els sistemes).
- Cost del kit d'avaluació: 457€ (1 per tots els sistemes).

➤ Cost total dels PCBs:

Els elements que requereixen PCB són tots excepte la placa de proves, que s'ha elaborat a través d'una placa de punts.

Es parteix d'un únic PCB en el qual es realitzaran el màxim nombre de circuits possible. Per la mida dels elements que hi constaran, s'estima que en 1 PCB hi cabrien els circuits de pràcticament dos sistemes:

- 2 unitats centrals.



- 1 programador d'EEPROM (únicament en cal un per a desenvolupar tots els sistemes).
- 4 nodes de 3 vibradors.
- 30 nodes d'un vibrador.

El cost d'aquest PCB oscil·la entorn dels 320€. Si es vol realitzar un únic sistema aquest cost ascendeix igualment als 320€.

➤ Cost total del muntatge:

El muntatge val uns 20€/placa. Per tant, el cost total és d'uns 380€. Per als següents sistemes no s'ha de repetir el muntatge del mòdul de programació de l'EEPROM.

Cal dir que la placa de proves que s'ha desenvolupat s'ha realitzat de manera manual, i es comptabilitzarà com a tal. El cost aproximat en un laboratori d'una placa amb aquestes característiques (per exemple al Laboratori Comú de Mecànica) seria d'uns 40€.

➤ Cost de la verificació:

El temps necessari per a la verificació de cada placa i posada a punt (emmagatzematge del seu primer codi) oscil·la entorn dels 15 minuts, a fi de comprovar que la memòria EEPROM s'hagi escrit correctament, que les connexions siguin correctes, assignar a cada unitat un primer codi i col·locar la corresponent bateria. Per tant, en unes 5 hores (uns 75€) s'haurà verificat el sistema.

Ara es realitzarà una estimació de costos de disseny i desenvolupament del sistema.

S'estimen les següents hores de realització:

- 250 hores d'avaluació d'alternatives per als diferents components i diverses proves amb components.
- 120 hores de disseny i elaboració de hardware.
- 1040 hores de disseny del software de les diferents unitats.
- 90 hores de disseny del software del PC.
- 200 hores de direcció de projecte.



Però una gran part d'aquestes hores han estat destinades a l'aprenentatge i formació en general (sobre els entorns de programació, el microprocessador en sí, etc.) i a la presa de contacte amb l'ensamblador. Si el projecte hagués estat elaborat per una empresa ja familiaritzada en el sector, s'estimarien les següents hores:

- 100 hores d'avaluació d'alternatives per als diferents components i diverses proves amb components.
- 90 hores de disseny i elaboració de hardware.
- 200 hores de disseny del software de les diferents unitats.
- 60 hores de disseny del software del PC.
- 50 hores de direcció de projecte.

Ponderant les hores com a direcció de projecte, analista i programador respectivament a 10%, 30%, 60%, i els corresponents costos per a l'empresa en cada càrrec com a 5000€/mes (sou de 3000€/mes), 4000€/mes (2300€/mes), 3000€/mes (1700€/mes), es té un cost total d'uns 11.000€ en recursos humans.

Suposant que les comandes es facin de manera independent (ja que caldria un PCB per sistema), els costos variables del sistema ascendeixen a 1360€ (sense IVA). Com a costos variables s'han comptabilitzat:

- Cost corresponent als components d'1 unitat central, 15 mòduls d'un vibrador, 2 mòduls de 3 vibradors.
- El cost d'un PCB que contingui tots aquests elements (s'han suposat comandes independents amb conseqüent independència en PCBs).
- El cost del muntatge d'1 unitat central, 15 mòduls d'un vibrador, 2 mòduls de 3 vibradors.
- El cost de la verificació corresponent a la verificació i posada a punt d'1 unitat central, 15 mòduls d'un vibrador, 2 mòduls de 3 vibradors.

Els costos fixos del sistema pel què respecta al hardware ascendeixen a 620€ (sense IVA). Pel què fa a costos d'amortització d'eines i d'infraestructures d'oficina s'han comptabilitzat uns 450€, de manera que, en total, els costos fixos ascendeixen a 1070€.



Així, el cost total d'un únic sistema és de 2450€ (sense IVA, sumant el cost fix i el variable). Cal dir que el cost d'aquest hardware és molt baix en comparació al d'altres sistemes utilitzats en realitat virtual, i encara es podria reduir més en cas de fabricació d'un nombre considerable de sistemes. És molt difícil trobar alternatives en el mercat de sistemes d'interacció implícita de menor cost, i a més, les característiques que presenten no són les que es desitgen per al client.

Sumant a aquesta quantitat la corresponent al disseny de 11.000€, la xifra total és de 13.430€ (sense IVA). La quantitat total incloent l'impost sobre el valor afegit és de 15.580€.

F.2. Terminis

A continuació es presentarà el primer Gantt que es va realitzar de cara a fer una primera previsió de terminis i camins crítics. Aquests vénen representats en vermell.

En ser la primera planificació, el resultat final no ha estat ben bé així. No s'hi va comptabilitzar el primer mes de projecte però en canvi es va comptar l'entrega del primer prototipus durant el mes de març, cosa que si es disposés ja dels recursos econòmics hagués estat cert, i s'hagués acomplert amb el termini, amb temps suficient per a fabricar els prototipus.

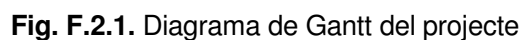
S'han posat en paral·lel les tasques de disseny de hardware i de software ja que es podien realitzar amb independència, i es van anar desenvolupant en paral·lel. Es va concebre aquesta distribució de manera que en paral·lel al desenvolupament del software es verificaven les diferents rutines. Per tant aquest és el camí crític ja que al final la verificació és l'última tasca a completar.

Si s'hagués de refer aquest diagrama, començaria per retardar la realització de la interfície amb el computador, ja que aquesta es va realitzar en paral·lel amb el software dels mòduls però cap al final del projecte, no es va fer a l'inici.

El PCB, el muntatge i la verificació del hardware en realitat s'hagués realitzat també al final del projecte, tot i que si hi hagués el pressupost necessari s'hagués pogut realitzar abans, tal i com estava contemplat en aquest primer Gantt.

Segons aquesta primera previsió el lliurament del primer prototipus es podria dur a terme durant el mes de març del 2005.





Representant ara el diagrama de Pert:

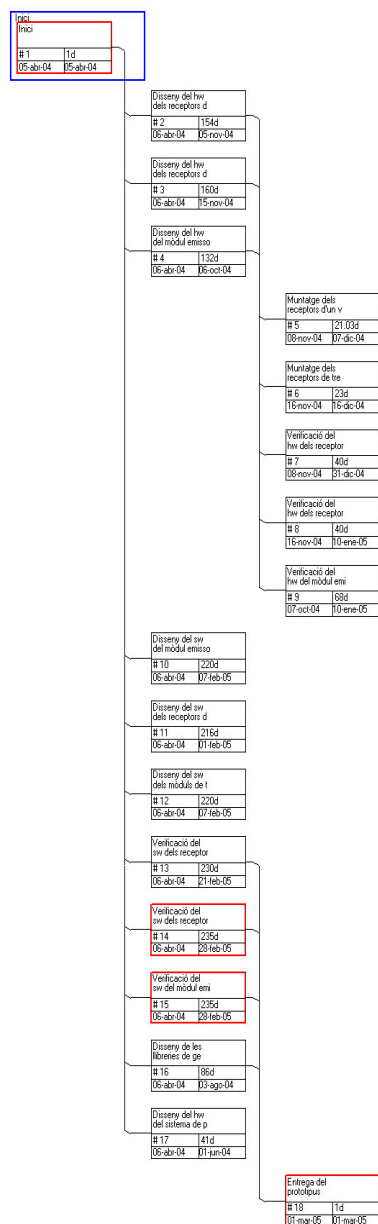


Fig. F.2.2. Diagrama de Pert del projecte



G. Càlcul de l'abast

G.1. Nocions bàsiques sobre radiofreqüència

La unitat de mesura més habitual en RF per descriure un senyal és la potència que emet una determinada càrrega.

$$P(t) = v(t) \cdot i(t) = \frac{v^2(t)}{R} = i^2(t) \cdot R \quad [W] \quad (\text{Eq. G.1.1.})$$

Ja que, segons la llei d'Ohm:

$$v(t) = i(t) \cdot R \quad (\text{Eq. G.1.2.})$$

La unitat de mesura de potència en el sistema internacional és el *Watt*, però en el cas de les comunicacions els nivells de potència són de l'ordre de mW o μW, de manera que s'acostuma a treballar amb magnituds logarítmiques com els dB. El dB es defineix com a una relació logarítmica entre dues potències P_1 i P_2 .

Per a establir la relació entre la potència i el voltatge cal utilitzar una impedància de referència sobre la qual definir la potència.

$$dB = 10 \log \left[\frac{P_1}{P_2} \right] = 10 \log(P_1) - 10 \log(P_2) \quad (\text{Eq. G.1.3.})$$

$$dB = 10 \log \left[\frac{P_1}{P_2} \right] = 10 \log \left[\frac{v_1^2 / R_1}{v_2^2 / R_2} \right] = 10 \log \left[\frac{v_1^2}{v_2^2} \right] + 10 \log \left[\frac{R_1}{R_2} \right] = 20 \log \left[\frac{v_1}{v_2} \right] + 10 \log \left[\frac{R_1}{R_2} \right]$$

(Eq. G.1.4.)

Si les dues impedàncies de referència són iguals,

$$dB = 10 \log \left[\frac{P_1}{P_2} \right] = 20 \log \left[\frac{v_1}{v_2} \right] \quad (\text{Eq. G.1.5.})$$

Si es pren com a potència de referència $P_2=1\text{mW}$, la relació en dB deixa de ser una magnitud relativa i passa a ser una magnitud absoluta de potència anomenada dBm.



$$dBm = 10 \log \left[\frac{P}{1mW} \right] \quad (\text{Eq. G.1.6.})$$

El mateix es pot fer prenent com a potència de referència 1W i la magnitud resultant són els dBW.

En cas que s'utilitzi una tensió de referència d'1μV:

$$dB\mu V = 20 \log \left[\frac{v}{1\mu V} \right] \quad (\text{Eq. G.1.7.})$$

Si es desitja aprofundir en aquest tema es pot recórrer al llibre *Radiofreqüència. Una introducció experimental* d'Edicions UPC, 1998.

L'antena que es té és de tipus *loop*. Aquestes antenes es poden elaborar mitjançant un cordó de coure en pla sobre el propi circuit imprès. Els paràmetres que defineixen les antenes d'aquest tipus són els mostrats en la figura G.1.1.

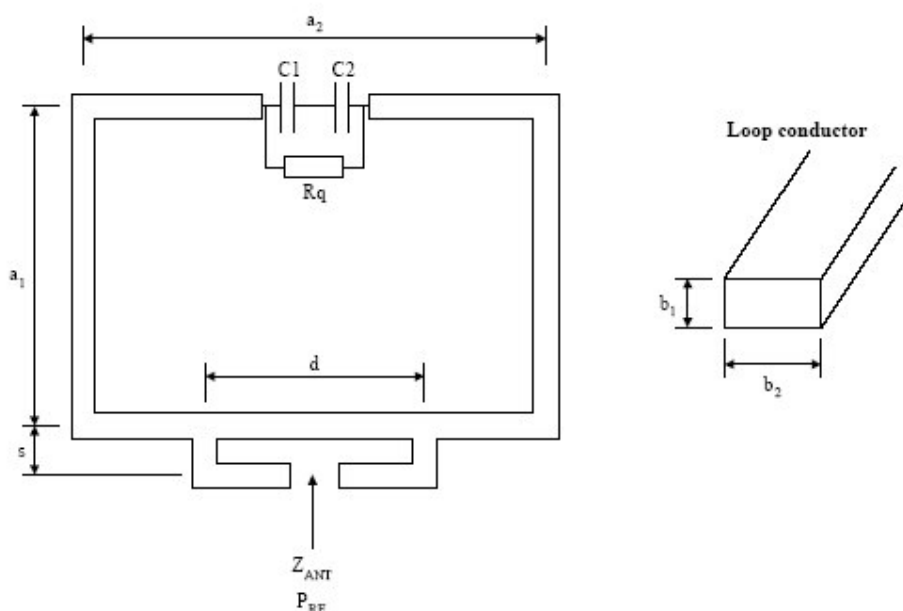


Fig. G.1.1. Paràmetres que defineixen l'antena *loop*

Els paràmetres físics de l'antena *loop* són:



- a_1 = amplada de l'antena [m].
- a_2 = longitud de l'antena [m].
- b_1 = gruix del conductor [m].
- b_2 = amplada del conductor [m].

Per al càlcul dels paràmetres elèctrics de l'antena el llaç rectangular s'ha de modelar com a una espira quadràtica equivalent, i el conductor pla ha de ser modelitzat com un conductor amb un radi equivalent. El costat quadràtic de l'espira haurà de ser doncs:

$$a = \sqrt{a_1 \cdot a_2} \quad [m] \quad (\text{Eq. G.1.8.})$$

L'àrea de l'espira serà:

$$A = a^2 \quad [m^2] \quad (\text{Eq. G.1.9.})$$

El radi equivalent del conductor de l'espira és aquell que té una capacitat igual a la de la geometria no circular.

$$b = 0.35 \cdot b_1 + 0.24 \cdot b_2 \quad [m] \quad (\text{Eq. G.1.10.})$$

A fi de poder estimar la capacitat C_p que fa entrar en ressonància l'antena s'ha de determinar la impedància d'entrada de l'espira. Així es podrà determinar l'eficiència de l'antena, la resistència a la radiació i pèrdues.

El circuit equivalent per l'impedància d'entrada de l'antena quan es troba en mode de transmissió és:

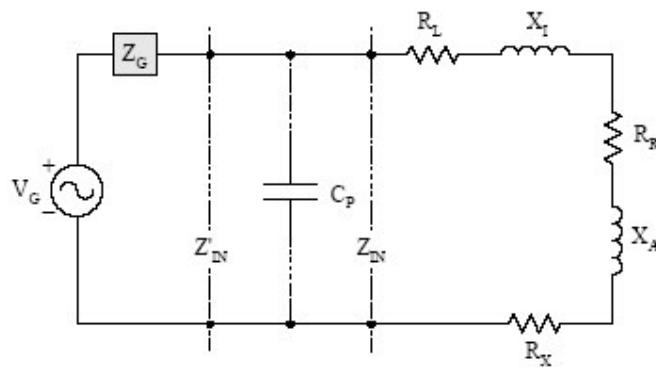


Fig. G.1.2. Circuit equivalent de l'antena en mode de transmissió



D'on la impedància equivalent per l'entrada és:

$$Z_{IN} = (R_R + R_L + R_X) + j \cdot 2 \cdot \pi \cdot f_0 \cdot (L_A + L_I) \quad [\Omega] \quad (\text{Eq. G.1.11.})$$

- R_R = Resistència a la radiació $[\Omega]$.
- R_L = Pèrdua de resistència del conductor *loop* $[\Omega]$.
- R_X = pèrdues addicionals $[\Omega]$.
- L_A = Inductància de l'antena *loop* $[H]$.
- L_I = Inductància del conductor de l'antena $[H]$.

La resistència a la radiació val:

$$R_R \approx 31171 \cdot \left(\frac{A^2}{\lambda^4} \right) \quad [\Omega] \quad (\text{Eq. G.1.12.})$$

- amb $\lambda = \frac{c}{f_0} \quad [m]$ (Eq. G.1.13.)
- $c = 3 \cdot 10^8 \text{ m/s}$ velocitat de la llum.
- f_0 és la freqüència de ressonància en Hz.

La resistència de pèrdua de l'inductor de l'antena *loop* val:

$$R_L = \frac{l}{P} \cdot R_s = \frac{a_1 + a_2}{b_1 + b_2} \cdot \sqrt{\frac{\pi \cdot f_0 \cdot \mu_0}{\sigma}} \quad [\Omega] \quad (\text{Eq. G.1.14.})$$

- l = longitud del conductor de metall del *loop*.
- P = perímetre de la secció en creu del conductor.
- R_s = resistència de la superfície del conductor.
- $\mu_0 = 4 \cdot \pi \cdot 10^{-7} \text{ H/m}$ permeabilitat magnètica del buit.
- $\sigma = 5,8 \cdot 10^7$ conductivitat del conductor, en concret del coure.



Les pèrdues addicionals són introduïdes principalment per la resistència en sèrie equivalent de la capacitat C_P , i venen donades per:

$$R_X = \frac{2 \cdot \pi \cdot f_0 \cdot (L_A + L_I)}{Q} - R_R - R_L \quad [\Omega] \quad (\text{Eq. G.1.15.})$$

Q és el factor de qualitat de l'antena i és principalment determinada per la resistència en sèrie equivalent de C_P . Una resistència en paral·lel R_Q es pot usar per a controlar el factor de qualitat de l'antena, reduint la impedància d'entrada de l'antena.

C_P s'usa en paral·lel a Z_{IN} per a fer entrar en ressonància l'antena, cancel·lant la part imaginària de la impedància d'entrada Z_{IN} a la freqüència d'operació. C_P i Z_{IN} en ressonància val:

$$C_P = \frac{L_A + L_I}{(R_R + R_L + R_X)^2 + [2 \cdot \pi \cdot f \cdot (L_A + L_I)]^2} \quad [F] \quad (\text{Eq. G.1.16.})$$

$$Z_{IN} = R_R + R_L + R_X + \frac{[2 \cdot \pi \cdot f \cdot (L_A + L_I)]^2}{R_R + R_L + R_X} \quad [\Omega] \quad (\text{Eq. G.1.17.})$$

On:

$$\triangleright L_A = 2 \cdot \mu_0 \cdot \frac{a}{\pi} \cdot \left[\ln\left(\frac{a}{b}\right) - 0.774 \right] \quad [H] \quad (\text{Eq. G.1.18.})$$

en el cas d'antena quadrada de costat a i radi b

$$\triangleright L_I = \mu_0 \cdot \frac{A}{2 \cdot a} \quad [H] \quad (\text{Eq. G.1.19.})$$

Així, l'eficiència de l'antena val:

$$\eta = \frac{R_R}{R_R + R_L + R_{ESR}} = \frac{Q \cdot R_R}{2 \cdot \pi \cdot f_0 \cdot (L_A + L_I)} \quad (\text{Eq. G.1.20.})$$

Respecte als càlculs de rang, els paràmetres bàsics són:

- P_{RF} [dBm] potència de transmissió.
- η eficiència de l'antena de l'emissor i del receptor.
- R [m] separació de l'antena (abast).



- L_p [dB] pèrdua en espai lliure.
- L_x [dB] pèrdues addicionals en la propagació.
- S [dBm] sensibilitat del receptor.

La propagació en espai lliure serveix per a determinar la força del senyal rebut quan l'emissor i el receptor no presenten cap obstacle entre ells, tenint únicament en compte les pèrdues degudes a la propagació esfèrica de l'energia de l'antena.

$$L_p = \left(\frac{\lambda}{4 \cdot \pi \cdot R} \right)^2 \quad (\text{Eq. G.1.21.})$$

$$L_p[\text{dB}] = 20 \cdot \log \left(\frac{\lambda}{4 \cdot \pi \cdot R} \right) \quad (\text{Eq. G.1.22.})$$

El rang de comunicació amb antenes iguals per a la transmissió i la recepció, sense corregir per pèrdues diverses (L_x) val:

$$R = \frac{\lambda}{4 \cdot \pi \cdot \sqrt{\frac{S}{\eta^2 \cdot P_{RF}}}} \quad [m] \quad (\text{Eq. G.1.23.})$$

Aquesta equació és vàlida per a $R > 2 \cdot D^2 / \lambda$, on D és la major dimensió de l'antena.

Donat un determinat rang R , pèrdues L_x , sensibilitat S i idèntiques antenes per a transmissió i recepció, la potència necessària P_{RF} val:

$$P_{RF} = \frac{S}{L_x \cdot L_t \cdot \eta^2} \quad [W] \quad (\text{Eq. G.1.24.})$$

G.2. Determinació de l'abast màxim

El fabricant estima que:

- En la banda de 868MHz, $\lambda \approx 0.35\text{m}$.
- Una antena 9.5mmx9.5mm té un guany típic de 20dB fins a -25dB.



- A 868MHz, 10dBm de potència de sortida i -105 de sensibilitat el guany és aproximadament de -20dB.
- També indica que les pèrdues típiques a 868MHz degudes a obstacles són:

Object causing path loss	Typical loss [dB]
Wall (indoor)	10-15
Wall (exterior)	2-38 (percentage of windows and height important)
Floor	12-27
Window	2-30 (metal tinted windows cause high loss)

Fig. G.2.1. Pèrdues típiques degudes a obstacles en la transmissió

Seguidament s'estimaran teòricament aquests paràmetres.

L'antena *loop* per a la banda de 868MHz s'ha dissenyat seguint estrictament les recomanacions del fabricant expressades en la figura G.2.2.

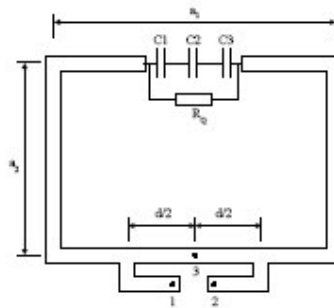


Fig. G.2.2. Circuit equivalent de l'antena en mode de transmissió

Aquesta antena permet treballar en la banda de 868MHz.

Els paràmetres que la descriuen són:

- a_1 longitud de l'antena [mm].
- a_2 amplada de l'antena [mm].



- d amplada de l'alimentació [mm].
- b amplada del conductor del *loop*.

L'antena està dissenyada per a tenir ressonància en 868 gràcies als condensadors C_1 , C_2 , C_3 .

A fi de mantenir les variacions degudes als condensadors en el guany de l'antena inferiors a 3dB, C_1 , C_2 , C_3 han de tenir toleràncies iguals o millors a $\pm 0.1\text{pF}$.

La resistència R_0 és la que determina el factor de qualitat de l'antena. El *loop* està regulat a aproximadament 300Ω . L'amplada del conductor és 1.524mm. Els paràmetres subministrats pel propi fabricant són els inclosos en la taula G.2.1.

Loop antenna $a_1 \times a_2$ [mm]	Feed length d [mm]	C1		C2		C3		R_0 [k Ω]	Q-value	Z_0 [Ω]
		Value [pF]	Tolerance [pF]	Value [pF]	Tolerance [pF]	Value [pF]	Tolerance [pF]			
9.5x9.5	13	3.9	± 0.1	4.7	± 0.1	5.6	± 0.1	18	40	300

Taula G.2.1. Paràmetres de l'antena *loop*

Altres factors que interessen en els càlculs són:

- La potència màxima de transmissió és de 10dBm, i és la que s'utilitza.
- La sensibilitat en la recepció és de -100dBm.
- La banda de treball és els 868MHz. El canal és 869.6MHz (canal 124).
- Gruix per sobre del substrat 35 μm .

Tant l'antena de la unitat central com les dels 2 tipus de nodes actuadors són iguals a la descrita; a més s'ha implementat per codi que tots els elements transmetin a màxima potència quan es trobin en emissió, de manera que es té un únic valor d'abast aproximat.

Per al càlcul dels paràmetres elèctrics de l'antena, s'ha dit que el llaç rectangular s'ha de modelar com a una espira quadràtica equivalent. El costat quadràtic és:

$$a = \sqrt{a_1 \cdot a_2} = \sqrt{9.5 \cdot 9.5} = 9.5 \quad [\text{mm}] \quad (\text{Eq. G.2.1.})$$



L'àrea de l'espira:

$$A = a^2 = 90.25 \quad [mm^2] \quad (\text{Eq. G.2.2.})$$

El radi equivalent del conductor:

$$b = 0.35 \cdot b_1 + 0.24 \cdot b_2 = 0.35 \cdot 0.035 + 0.24 \cdot 1.524 = 0.37801 \quad [mm] \quad (\text{Eq. G.2.3.})$$

Ara es trobaran els diferents paràmetres que configuren el circuit equivalent:

$$\text{Prèviament cal determinar el valor de } \lambda = \frac{c}{f_0} = \frac{3 \cdot 10^8}{869.6 \cdot 10^6} = 0.345 [m] \quad (\text{Eq. G.2.4.})$$

Resistència a la radiació:

$$R_R \approx 31171 \cdot \left(\frac{A^2}{\lambda^4} \right) \quad [\Omega] = 31171 \cdot \left(\frac{0.00009025^2}{0.345^4} \right) = 0.01792 \Omega \quad (\text{Eq. G.2.5.})$$

Pèrdua de resistència del conductor loop:

$$R_L = \frac{l}{P} \cdot R_s = \frac{a_1 + a_2}{b_1 + b_2} \cdot \sqrt{\frac{\pi \cdot f_0 \cdot \mu_0}{\sigma}} = \frac{0.0095 + 0.0095}{0.001524 + 0.000035} \cdot \sqrt{\frac{\pi \cdot 869.6 \cdot 10^6 \cdot 4 \cdot \pi \cdot 10^{-7}}{5.8 \cdot 10^7}} = 0.09376 \Omega \quad (\text{Eq. G.2.6.})$$

Inductància de l'antena loop:

$$L_A = 2 \cdot \mu_0 \cdot \frac{a}{\pi} \left[\ln \left(\frac{a}{b} \right) - 0.774 \right] = 2 \cdot 4 \cdot \pi \cdot 10^{-7} \cdot \frac{0.0095}{\pi} \left[\ln \left(\frac{0.0095}{0.00037801} \right) - 0.774 \right] = 1.862 \cdot 10^{-8} H \quad (\text{Eq. G.2.7.})$$

Inductància del conductor de l'antena:

$$L_I = \mu_0 \cdot \frac{A}{2a} = 4 \cdot \pi \cdot 10^{-7} \cdot \frac{0.00009025}{2 \cdot 0.0095} = 5.969 \cdot 10^{-9} H \quad (\text{Eq. G.2.8.})$$

Pèrdues addicionals:

$$R_X = \frac{2 \cdot \pi \cdot f_0 \cdot (L_A + L_I)}{Q} - R_R - R_L =$$



$$= \frac{2 \cdot \pi \cdot 869.6 \cdot 10^6 \cdot (1.862 \cdot 10^{-8} + 5.969 \cdot 10^{-9})}{40} - 0.01792 - 0.09376 = 3.2471 \Omega \quad (\text{Eq. G.2.9.})$$

D'on la impedància equivalent per l'entrada és:

$$Z_{IN} = (R_R + R_L + R_X) + j \cdot 2 \cdot \pi \cdot f_0 \cdot (L_A + L_I) = 3.3588 + 134.3508j \quad \Omega \quad (\text{Eq. G.2.10.})$$

C_P i Z_{IN} en ressonància valen:

$$C_P = \frac{L_A + L_I}{(R_R + R_L + R_X)^2 + [2 \cdot \pi \cdot f \cdot (L_A + L_I)]^2} = 1.3614 \cdot 10^{-12} F \quad (\text{Eq. G.2.11.})$$

$$Z_{IN} = R_R + R_L + R_X + \frac{[2 \cdot \pi \cdot f \cdot (L_A + L_I)]^2}{R_R + R_L + R_X} = 5377.343 \quad \Omega \quad (\text{Eq. G.2.12.})$$

Així, l'eficiència de l'antena val:

$$\eta = \frac{R_R}{R_R + R_L + R_X} = \frac{Q \cdot R_R}{2 \cdot \pi \cdot f_0 \cdot (L_A + L_I)} = \frac{40 \cdot 0.01792}{134.3508} = 0.0053353 = -22.72 dB \quad (\text{Eq. G.2.13.})$$

Cal notar que aquest valor s'aproxima molt al guany de -20dB donat pel fabricant.

Respecte als càlculs de rang, si $S = -100 dBm = 10^{-13} W$ és la sensibilitat del receptor, η l'eficiència i $P_{RF} = 10 dBm = 0.01 W$ la potència de transmissió, l'abast màxim val:

$$R = \frac{\lambda}{4 \cdot \pi \cdot \sqrt{\frac{S}{\eta^2 \cdot P_{RF}}}} = \frac{0.345}{4 \cdot \pi \cdot \sqrt{\frac{10^{-13}}{0.0053353^2 \cdot 10 \cdot 10^{-3}}}} = 46.32 \quad [m] \quad (\text{Eq. G.2.14.})$$



H.2. Esquemàtic dels nodes actuadors d'un vibrador

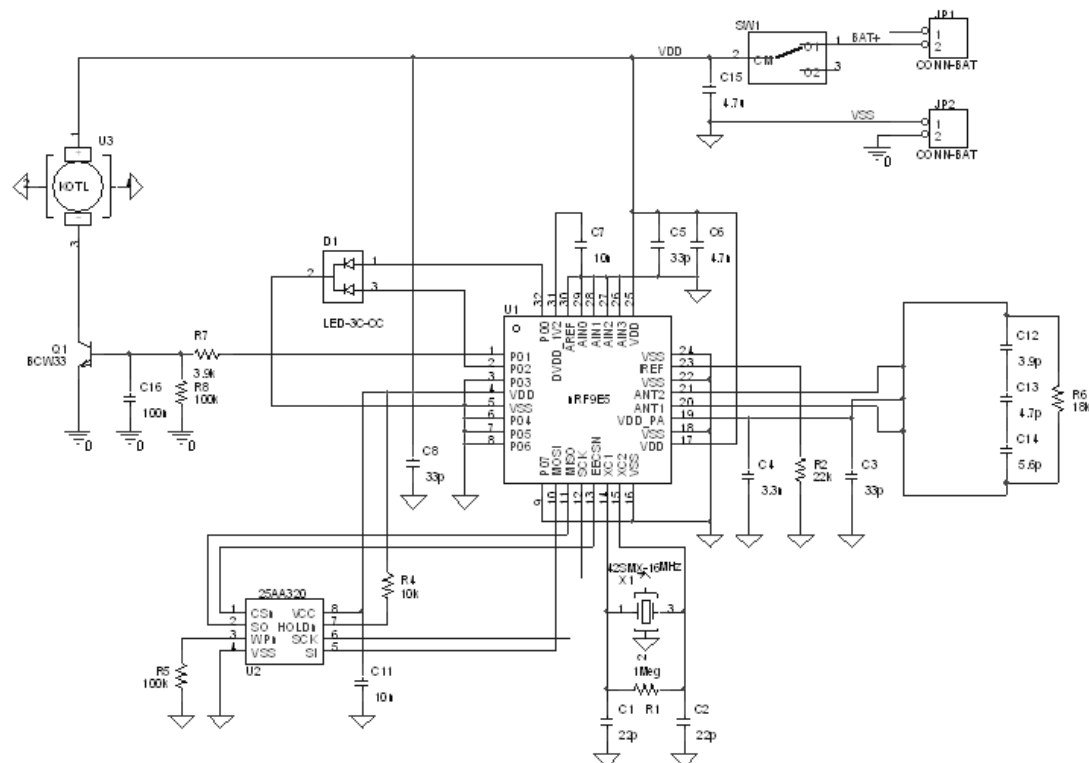


Fig. H.2.1. Disseny en Orcad® dels nodes d'un vibrador



H.3. Esquemàtic dels nodes actuadors de tres vibradors

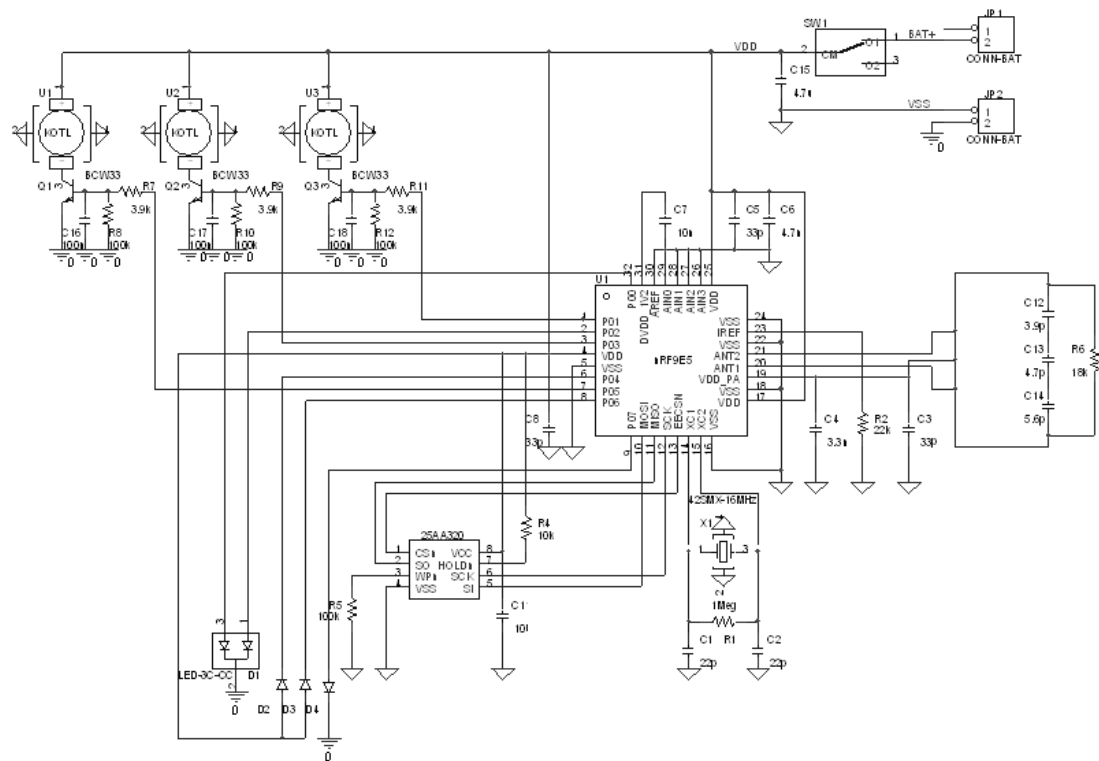


Fig. H.3.1. Disseny en Orcad ® dels nodes de 3 vibradors



H.4. Esquemàtic del mòdul de programació de l'EEPROM

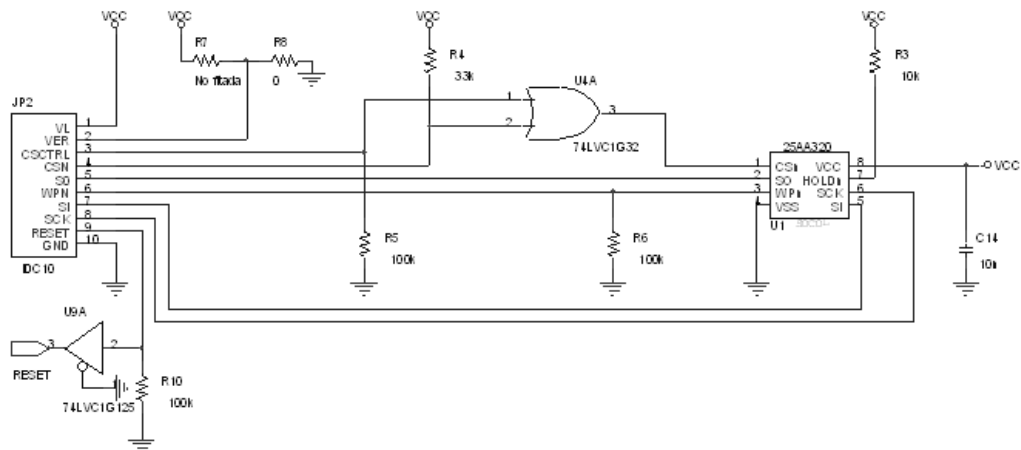
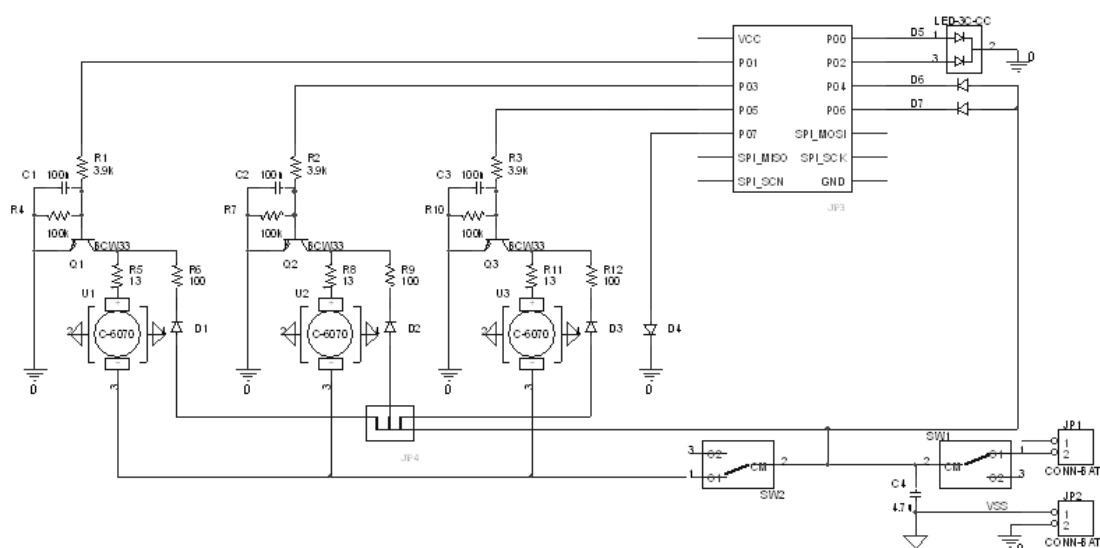


Fig. H.4.1. Disseny en Orcad ® del programador de l'EEPROM







I. Codi implementat en la unitat central

```

;= unitat central.a51 =====
;Software de control de la unitat central, usant el timer 1 per al bus sèrie.
;
;El RTC s'usa per a generar la interrupció periòdica per a empaquetar les dades que s'han anat rebent per UART, formant
el ;missatge que el mòdul haurà d'enviar als mòduls actuadors tal i com l'esperen. Aquest enviament es farà cada 100ms.
;L'estat dels nodes es mantindrà a l'últim valor mentre no arribi per UART una comanda d'estat que indiqui una variació
en ;algun dels nodes del sistema.
;Aquesta actualització de l'estat dels actuadors no es realitzarà en cas que es rebí pel port sèrie una de les següents
comandes:
;-una comanda de petició de identificació (amb un únic receptor en funcionament)
;-una comanda de petició d'estat i nivell de bateria d'un node
;-una comanda per a fixar un nou codi identificador (un únic receptor en funcionament)
;En ser aquestes comandes molt puntuals, s'ha decidit que un cop se'n rebí una per UART s'esperarà a la següent
interrupció ;del RTC a enviar-la (cada 100 ms) i en aquesta interrupció no s'actualitzarà l'estat dels actuadors. Així no és
necessari ;augmentar el temps de recepció dels nodes, per la qual cosa es té un estalvi en bateria.
;En els tres casos s'esperarà un byte de resposta per part del mòdul actuator, el qual contindrà informació diferent segons
la ;comanda. Aquest byte es tractarà al PC un cop li hagi arribat pel port sèrie. Així, la unitat central enviarà el byte de
resposta ;al PC sense cap modificació. L'espera de la resposta de l'actuator s'ha implementat a fi que no sigui activa, sinó
mitjançant ;l'ús d'interrupcions. En cas que el missatge rebut per UART sigui d'actualització de l'estat d'un node es fa un
eco immediat de ;la comanda, sense espera de resposta per part del receptor.
;Es forçarà periòdicament una resincronització dels rellores entre els receptors i l'emissor, a fi de minimitzar la pèrdua de
;missatges.
;Es disposarà d'un led connectat a l'alimentació general que romandrà encés mentre el mòdul es trobi en connexió, ja que
no hi ha restriccions d'alimentació. Dos LEDs més complementaris indiquen quan s'estan enviant trames per
radiofreqüència i de ;quin tipus. Un dels LEDs s'il·lumina cada cop que s'envia un missatge d'estat, i l'altre s'il·lumina cada
cop que s'efectua un ;enviament d'un missatge de comanda.
;=====

$nomod51
$include(nRF9E5.inc)                                ;assignacions dels SFR del nRF9E5

bdata                                                ;espai de memòria adreçable a nivell de bit

oper:        ds      1                                ;nou estat de l'actuator a enviar per RF
serieIn:      ds      1                                ;informació rebuda des del PC
serieOut:     ds      1                                ;informació a enviar al PC
misComRF:    ds      1                                ;informació de comanda per a enviar per RF
misRebRF:    ds      1                                ;informació de resposta a comanda rebuda per RF

data                                                  ;espai de memòria adreçable a nivell de byte

cont:        ds      1                                ;comptador per a la transmissió de bytes

```



```

misEstRF:      ds      8          ;informació estat ja empaquetada per a enviar per RF
nIntents:      ds      1          ;comptador nombre de repeticions en cas de no obtenir resposta

data          at        0x20

flags:         ds      1
flagBat        bit     flags.0    ;flagBat=1 si si s'espera resposta de bateria/estat
flagResp       bit     flags.1    ;flagResp=1 si s'ha rebut una resposta d'un actuator
;flagLBT       bit     flags.2    ;això s'usaria en cas que es volgués protocol LBT

idata         at        0x80

stack:                                     ;stack

code at 0      ;vectors d'interrupcio (inici)
               ljmp    inici
code at 0x23    ;vector d'interrupcio (RI o TI port serie)
               ljmp    siSerie
code at 0x53    ;vectors d'interrupcio (transceiver RF)
               ljmp    siRF
code at 0x63    ;vectors d'interrupcio (RTC)
               ljmp    siRTC
code           ;codi reubicable

;programa principal (RTC inicialment aturat i a 0)

inici:         mov     SP,#stack-1    ;inicialització del stack pointer
               mov     SPICLK,#0x00   ;SPICLK configurat a CPU_CLK/2
               mov     SPI_CTRL,#0x02 ;SPI connectat al transceiver RF
               clr     SBSCN          ;comença comanda en bus SPI
               mov     a,#0x09        ;a=comanda d'escriptura a partir de
                                   ;registre de configuracio 9 (CRC & CLK)
               acall    spi_wr        ;trameta comanda
               mov     a,#0xdf        ;a=dada per registre 9 (CRC & CLK)
               acall    spi_wr        ;trameta dada (2 bytes CRC, cristall=16 MHz, CPU_CLK=cristall)
               setb     SBSCN         ;fi comanda en bus SPI
               mov     CKLFCN,#0x1f   ;cristall=16 MHz i CPU_CLK=cristall. Queda garantit CKLF de 4 kHz
               mov     TICK_DV,#0x01  ;queda garantit TICK de 0,5 ms
               mov     P0_ALT,#0x06   ;P0 configurat com a E/S generals, P0.1 com a RXD, P0.2 TXD
               mov     P0_DIR,#0xaa   ;P00,P04,P06 configurats com a sortides
                                   ;P03,P05,P07 configurats com a entrades
                                   ;P01 RXD in, P02 TXD out usarem full duplex
               mov     P0_DRV,#0xaa   ;P0: bits parells amb corrents elevats
               mov     P0,#0xff       ;apaga tots els led
               acall    initRF        ;crida a la inicialització comunicació RF
               acall    initUart      ;crida a la inicialització comunicació UART

```



```

mov IE,#0x10      ;activa global interrupt i del port serie (RI o TI)
mov IP,#0x10      ;port sèrie alta prioritat,prioritat natural=5 (RTC 12)
acall initMis      ;inicialitza el valor del missatge periòdic als actuadors
mov misComRF,#0x00 ;inicialitza el valor del missatge de comanda als actuadors
mov cont,#0x00     ;inicialitzem el contador de #bytes enviats per RF
mov nIntents,#0x00 ;inicialitzem el comptador de nombre d'intents
clr flagBat        ;inicialitzem el flag d'espera de lectura de bateria a 0
clr flagResp       ;inicialitzem el flag de resposta rebuda a 0
clr TX_EN          ;selecció del transceiver en recepció
setb TRX_CE        ;habilita el transceiver
llac0: mov a,REGX_CTRL ;a=REGX_CTRL
jnb ACC.4,llac0     ;espera fins que REGX_CTRL.4 sigui 0
mov REGX_MSB,#0
mov REGX_LSB,#200-1
mov REGX_CTRL,#0x0a;defineix periodicitat RTC a 100 ms i l'engega
                        ;nomes cal definir 16 LSB (8 MSB a 0 per defecte)
setb EWDI          ;habilita interrupcio wakeup (RTC)
setb EX4           ;habilita la interrupció del transceiver RF
setb EA            ;habilita interrupcions (global)
llac1: mov a,REGX_CTRL ;a=REGX_CTRL
jnb ACC.4,llac1     ;espera fins que REGX_CTRL.4 sigui 0,interfaç lliure
llac2: mov PCON,#0xB1 ;força el mode idle
sjmp llac2          ;pas a mode idle:la MCU es podrà despertar per qualsevol
                        ;de les interrupcions actives o per watchdog reset

```

;servei d'interrupció del RTC

```

siRTC:  clr WDTI      ;esborra la interrupció per RTC
        push acc
        push b
        push psw
        clr SBCCSN    ;comença comanda en bus SPI
        mov a,#0x20   ;a=comanda d'escriptura de les dades d'un
                        ;missatge (escriu a TX_payload des del byte 0)
        acall spi_wr   ;tramet comanda
        mov a,misComRF
        jz tranMis     ;si no hi ha comanda a transmetre actualitzem estat
tranCom: clr P04
        mov a,#0x08
        subb a,cont
        jz tranCom1
        mov a,misComRF ;a=byte a enviar=misComRF 8 cops
        acall spi_wr   ;tramet dada
        inc cont
        sjmp tranCom   ;salt a l'enviament del següent byte
tranCom1: setb SBCCSN  ;fi comanda en bus SPI
        setb TX_EN     ;selecciona el transceiver en emissió

```



```

        mov     cont,#0x00
        clr     EA                ;deshabilita interrupcions (global)
        clr     EX4
;si volguéssim aplicar CD en la transmissió dels missatges de comanda
;
        clr     TX_EN            ;selecciona el transceiver en recepció
;label1:      mov     c,CD        ;implementació d'un protocol LBT
;
        mov     flagLBT,c
;
        jc      label1           ;espera que CD=0 (quan el canal estigui buit)
;
        setb    TX_EN            ;selecciona el transceiver en transmissió
tranCom2:     jb      DR,tranCom2 ;espera transmissió preàmbul
tranCom3:     jnb     DR,tranCom3 ;espera transmissió paquet
        clr     TX_EN            ;selecciona el transceiver en recepció
        mov     c,flagResp
        djnz    nIntents,noReinici
        mov     misComRF,#0x00 ;inicialitza el valor del missatge de comanda als receptors
                                ;puc inicialitzar misComRF ja que no es tindrà una nova comanda
                                ;per part de l'usuari fins al cap de 100ms(mínim)per requeriment
noReinici:    setb     P04
        mov     a,EXIF           ;a=flags interrupcions esteses
        clr     ACC.6            ;esborra interrupcio transceiver de RF
        mov     EXIF,a          ;actualitza flags interrupcions esteses
        setb    EA              ;habilita interrupcions (global)
        setb    EX4
        sjmp    fi
tranMis:      clr     P06
        mov     a,#0x08
        subb    a,cont
        jz      tranMis1
        mov     a,cont
        add     a,#misEstRF
        mov     r0,a
        mov     a,@r0            ;a=byte a enviar
        acall   spi_wr          ;tramet dada
        inc     cont
        sjmp    tranMis         ;salt a l'enviament del següent byte
tranMis1:     setb    SBCCSN     ;fi comanda en bus SPI
        mov     cont,#0x00
        setb    TX_EN           ;transceiver en emissió
        clr     EA              ;deshabilita interrupcions (global)
        clr     EX4
tranMis2:     jb      DR,tranMis2 ;espera transmissió preàmbul
tranMis3:     jnb     DR,tranMis3 ;espera transmissió paquet
        clr     TX_EN           ;transceiver en recepció
        setb    P06
        setb    EA              ;habilita interrupcions (global)
        setb    EX4

```



```

                mov    a,EXIF      ;a=flags interrupcions esteses
                clr    ACC.6      ;esborra interrupcio transceiver de RF
                mov    EXIF,a     ;actualitza flags interrupcions esteses
fi:             pop    psw
                pop    b
                pop    acc
                reti

```

;servei d'interrupció del port serie

;nota:en cas de missatges que no siguin d'estat sinó de comanda d'usuari(nou codi, petició d'estat o d'identificació) suposem ;temps entre comandes>100ms de la interrupció, sinó es sobrescriuria la primera comanda. Per tant, com a requeriment per ;l'usuari, un cop ha introduït una d'aquestes comandes s'haurà d'esperar a veure una resposta en el PC abans d'introduir la ;següent.En cas de canviar 2 cops l'estat d'un cert mòdul en l'interval entre interrupcions no s'apreciaran aquests canvis fins ;la següent interrupció del RTC

```

siSerie:        push    acc
                push    b
                push    psw
                jnb     TI,siRecepcio
                clr     TI
                pop     psw
                pop     b
                pop     acc
                reti

siRecepcio:     mov     serieIn,SBUF      ;serieIn=ABCD EFGH, FGH codi operació, ABCDE #mòdul
                mov     a,serieIn        ;en primer lloc comprovem que el missatge no conté un
                mov     c,P              ;bit erroni mitjançant la paritat. Si RB8 i P coincideixen,
                clr     a                ;o no hi ha cap error o un nombre parell d'errors (molt poc
                mov     acc.0,c          ;probable si els ports tenen la mateixa configuració)
                mov     b,a
                clr     a
                mov     c,RB8
                mov     acc.0,c
                clr     RI
                clr     P00
                cjne     a,b,noPar        ;si RB8!=P retornem un 0xFF (error de paritat)
                mov     a,serieIn        ;a=ABCD EFGH
                anl     a,#0xF8          ;a=ABCD E000
                cjne     a,#0x00,codiNoNul ;salta si ABCDE és no nul
                mov     a,serieIn        ;a=ABCD EFGH
                anl     a,#0x07          ;a=0000 0111
                subb     a,#0x01
                jz       codiNoNul       ;si serieIn=0000 0001 tractem el missatge normalment
                subb     a,#0x06         ;ja que el camp del codi no s'usa per res
                jz       codiNoNul       ;si serieIn=0000 0111 tractem el missatge normalment
                mov     a,#0xef          ;indicarem error de codi nul
                mov     c,P              ;paritat del byte emmagatzemat en l'acumulador

```



```

    mov    TB8,c          ;per transmetre el 9è bit de paritat
    mov    SBUF,#0xef     ;si reben una ordre per al mòdul 0 es torna per sèrie un 0xEF
    jmp    fiSerie        ;ja que només tenim codis de 1 a 31 i no es fa res
codiNoNul:
    jnb    serieIn.0,estat ;si el LSB val 0, es tracta d'un missatge d'estat
    clr    flagResp
    clr    flagBat
    mov    a,serieIn
    mov    misComRF,a
    anl    a,#0x07        ;a=0000 0FGH
    cjne   a,#0x05,noLectBat
    setb   flagBat        ;esperarem una resposta que contingui una lectura de bateria
noLectBat:
    mov    nIntents,#20d
    cjne   a,#0x07,fiSerie
    mov    a,#0x07        ;si SBUF acaba en 111 tornem 0x07 (error operació no existent)
    mov    c,P            ;paritat del byte emmagatzemat en l'acumulador
    mov    TB8,c          ;per transmetre el 9è bit de paritat
    mov    SBUF,#0x07
    mov    misComRF,#0x00 ;no transmetem aquesta comanda (comanda no existent)
    sjmp   fiSerie
noPar:
    mov    a,#0xff
    mov    c,P            ;paritat del byte emmagatzemat en l'acumulador
    mov    TB8,c          ;per transmetre el 9è bit de paritat
    mov    SBUF,#0xff     ;si el missatge es rep malament per UART torna 0xff
    sjmp   fiSerie
estat:
    mov    misComRF,#0x00 ;si arriba un missatge d'estat deixem d'enviar els de comanda
    mov    a,serieIn      ;a=ABCD EFG0
    mov    c,P            ;paritat del byte emmagatzemat en l'acumulador
    mov    TB8,c          ;per transmetre el 9è bit de paritat
    mov    SBUF,serieIn   ;SBUF=ABCD EFG0 es fa un eco cap al port serie
    anl    a,#0xE0        ;a=ABC0 0000
    swap   a              ;a=0000 ABC0
    rr     a              ;a=0000 0ABC=#byte
    add    a,#misEstRF
    mov    r1,a           ;r1=#misEstRF+#byte
    mov    a,serieIn
    anl    a,#0x06        ;a=0000 0FG0
    rr     a
    mov    oper,a         ;oper=0000 00FG
    mov    a,serieIn
    anl    a,#0x18        ;a=000D E000
    rr     a
    rr     a
    rr     a              ;a=0000 00DE, a indica el # de bit dins del byte
    jz     bit0
    dec    a
    jz     bit1

```



```

                dec    a
                jz     bit2
                sjmp   bit3
bit0:           mov    a,@r1        ;a=byte a enviar=misEstRF+nbyte
                mov    c,oper.0
                mov    acc.0,c
                mov    c,oper.1
                mov    acc.1,c
                mov    @r1,a        ;misEstRF+nbyte=a
                sjmp   fiSerie
bit1:           mov    a,@r1        ;a=byte a enviar=misEstRF+nbyte
                mov    c,oper.0
                mov    acc.2,c
                mov    c,oper.1
                mov    acc.3,c
                mov    @r1,a        ;misEstRF+nbyte=a
                sjmp   fiSerie
bit2:           mov    a,@r1        ;a=byte a enviar=misEstRF+nbyte
                mov    c,oper.0
                mov    acc.4,c
                mov    c,oper.1
                mov    acc.5,c
                mov    @r1,a        ;misEstRF+nbyte=a
                sjmp   fiSerie
bit3:           mov    a,@r1        ;a=byte a enviar=misEstRF+nbyte
                mov    c,oper.0
                mov    acc.6,c
                mov    c,oper.1
                mov    acc.7,c
                mov    @r1,a        ;misEstRF+nbyte=a
fiSerie:        setb   P00
                pop    psw
                pop    b
                pop    acc
                reti

```

;servei d'interrupció del transceiver de RF

;si rep una resposta per part d'algun dels nodes receptors envia per port sèrie el missatge rebut

```

siRF:          push   acc
                push   b
                push   psw
                mov    a,EXIF        ;a=flags interrupcions esteses
                clr    ACC.6        ;esborra interrupcio transceiver de RF
                mov    EXIF,a        ;actualitza flags interrupcions esteses
                clr    SBSCSN        ;comença comanda en bus SPI
                mov    a,#0x24        ;a=comanda de lectura de les dades d'un

```



```

;missatge (nombre de dades a RX_PW)
acall    spi_wr      ;tramet comanda
acall    spi_rd      ;s'espera un payload d'un byte
setb     SBCSN       ;fi comanda en bus SPI
mov      misRebRF,a
mov      serieOut,a  ;guardem a serieOut la resposta del receptor
mov      a,EXIF      ;a=flags interrupcions esteses
clr      ACC.6       ;esborra interrupcio transceiver de RF
mov      EXIF,a      ;actualitza flags interrupcions esteses
mov      a,serieOut
mov      c,P         ;paritat del byte emmagatzemat en l'acumulador
mov      TB8,c       ;per transmetre el 9è bit de paritat
mov      SBUF,serieOut ;enviament del missatge rebut per RF al port sèrie del PC
;tal i com es rep, sense tractar. El tractament es farà al PC

mov      c,flagBat
jc       compLectBat
mov      a,misRebRF
anl      a,#0x07
dec      a
jz       compLectCod
dec      a
dec      a
jz       compNouCod
sjmp     fiRF        ;si no és cap dels casos anteriors no s'accepta la resposta
compLectCod: sjmp     actFlagResp
compLectBat: clr      flagBat
sjmp     actFlagResp
compNouCod: mov      a,misRebRF
mov      b,misComRF
cjne     a,b,nofiRF
actFlagResp: setb     flagResp ;s'ha rebut una resposta, no un error
clr      flagBat ;ja no s'espera cap lectura de bateria
mov      nIntents,#0 ;no es torna a enviar la comanda
mov      misComRF,#0x00
sjmp     fiRF
nofiRF:   clr      flagResp
fiRF:     pop      psw
pop      b
pop      acc
reti

```

;subrutina initmis

;inicialitza el valor del missatge a enviar cada 100 ms sobre l'estat dels vibradors

;recursos: psw

; 1 nivell de stack




```

initMis:      mov     misEstRF ,#0x00  ;misEstRF inclou l'estat dels vibradors des de l'1
              mov     misEstRF+1,#0x00 ;al 31. Els dos bits corresponents a l'estat del
              mov     misEstRF+2,#0x00 ;vibrador 0 són els que indiquen el tipus de missatge
              mov     misEstRF+3,#0x00 ;rebut (en aquest cas és d'estat, i per tant valen 00).
              mov     misEstRF+4,#0x00
              mov     misEstRF+5,#0x00
              mov     misEstRF+6,#0x00
              mov     misEstRF+7,#0x00
              ret

```

;subrutina initRF

;inicialitza els paràmetres relatius a les transmissions per RF

```

;recursos:      psw
;               1 nivell de stack

```

```

initRF:         clr     SBCSN          ;comença comanda en bus SPI
              mov     a,#0x00          ;a=comanda d'escriptura a partir de
              ;registre de configuracio 0 (CH_NO)
              acall    spi_wr          ;tramet comanda
              mov     a,#0x7C          ;a=dada per registre 0 (CH_NO)
              acall    spi_wr          ;tramet dada (canal 124 => 869,6 MHz)
              mov     a,#0x0e          ;a=dada per registre 1 (varis)
              acall    spi_wr          ;tramet dada (potencia transmissio maxima,
              ;no retransmissio, recepcio normal i 868/915 MHz)
              mov     a,#0x44          ;a=dada per registre 2 (TX_RX_AFW)
              acall    spi_wr          ;tramet dada (adreces de 4 bytes)
              mov     a,#0x01          ;a=dada per registre 3 (RX_PW)
              acall    spi_wr          ;tramet dada (RX payload de 1 byte)
              mov     a,#0x08          ;a=dada per registre 4 (TX_PW)
              acall    spi_wr          ;tramet dada (TX payload de 8 bytes)
              mov     a,#0xA5          ;a=dada per registre 5 (RX_ADDRESS byte 0)
              acall    spi_wr          ;tramet dada
              mov     a,#0xED          ;a=dada per registre 6 (RX_ADDRESS byte 1)
              acall    spi_wr          ;tramet dada
              mov     a,#0xE6          ;a=dada per registre 7 (RX_ADDRESS byte 2)
              acall    spi_wr          ;tramet dada
              mov     a,#0xDE          ;a=dada per registre 8 (RX_ADDRESS byte 3)
              acall    spi_wr          ;tramet dada
              setb     SBCSN          ;fi comanda en bus SPI
              clr     SBCSN
              mov     a,#0x22          ;a=comanda d'escriptura de TX_ADDRESS des de byte 0
              acall    spi_wr          ;tramet comanda
              mov     a,#0xA5          ;a=dada per a byte 0 (TX_ADDRESS byte 0)
              acall    spi_wr          ;tramet dada
              mov     a,#0xED          ;a=dada per a byte 1 (TX_ADDRESS byte 1)
              acall    spi_wr          ;tramet dada
              mov     a,#0xE6          ;a=dada per a byte 2 (TX_ADDRESS byte 2)

```



```

acall    spi_wr      ;tramet dada
mov     a,#0xDE     ;a=dada per a byte 3 (TX_ADDRESS byte 3)
acall    spi_wr      ;tramet dada
setb    SBCSN       ;fi comanda en bus SPI
ret

```

;subrutina initUart

;inicialitza els paràmetres relatius a les transmissions per port sèrie UART

```

;recursos:      psw
;               1 nivell de stack

```

```

initUart:      mov    TH1,#243      ;BR=19.2kb/s si CPU_CLK=16MHz,SMOD=1,TIMER=1
               mov    CKCON,#0x10   ;TIM=1 (/4 timer clock)
               mov    PCON,#0x80    ;vull PCON.7=1=SMOD
               mov    SCON,#0xD2    ;volem serial mode 3, enable receiver, 9è bit com a bit de paritat
               mov    TMOD,#0x20
               mov    TCON,#0x40    ;inici del timer 1. Configuració per defecte de TMOD i T2CON a 0x00
               ret                 ;timer1 a mode 0(13-bit counter) i s'usa per a rx i tx.

```

;subrutina spi_rd: rep un byte del bus SPI. Par. sortida: a=byte rebut

```

;recursos:      psw
;               1 nivell de stack

```

;subrutina spi_wr: envia un byte pel bus SPI. Par. entrada: a=byte a enviar

```

;recursos:      a,psw
;               1 nivell de stack

```

```

spi_rd:
spi_wr:      anl     EXIF,#0xdf      ;esborra interrupcio SPI
               mov    SPI_DATA,a     ;SPI_DATA=dada a enviar (no caldria a spi_rd)
spi_rdwrl:   mov     a,EXIF           ;llegeix SPI_RX (EXIF.5)
               jnb    ACC.5,spi_rdwrl ;espera fins que SPI_RX=1
               mov    a,SPI_DATA     ;a=SPI_DATA (no caldria a spi_wr)
               ret
end

```



J. Codi implementat en el node actuador d'un vibrador

```

;= nodes actuadors d'un vibrador.a51 =====
;-cada 100 ms mira si ha rebut un missatge de RF (usa la interrupció 0x53 de RF)
;-resincronització cada 256 trames
;-llegeix l'estat de la bateria i el guarda a node.7(node.7=0 si Vsupply>=2.1,node.7=1 si V<2.1)
;LEDS:
;-si tots els vibradors parats:intermitència 0.1s on, 9.9 off a P02 si Vsupply és alt (en verd) i a P00 si Vsupply és baix (en
;roig), és a dir, segons node.7.
;-si algun vibrador en on: el led estarà 100ms encés,400ms parat segons Vsupply(node.7)
;-si comanda nova identificació completada:P00 i P02 0.1s en on a la vegada, 9.9 off
;-si comanda petició identificació:el led s'il·luminarà 500ms segons Vsupply (node.7)
;-si comanda petició estat i bateria:el led s'il·luminarà 500ms segons Vsupply (node.7)
;MOTORS:
;-mode 00: repòs. Llavors estatVib = 0 (tots els motors parats)
;-mode 01: 100ms vibrant cada 500 ms (f=2Hz, D=20%). estatVib =1 (algun motor en mode no nul)
;-mode 10: 100ms vibrant cada 200 ms (f=5Hz, D=50%). estatVib =1 (algun motor en mode no nul)
;-mode 11: 20ms vibrant cada 40 ms (f=25Hz, D=50%). estatVib =1 (algun motor en mode no nul)
;=====

$nomod51
#include(nRF9E5.inc)                                ;assignacions dels SFR del nRF9E5

data                                                ;espai de memòria adreçable a nivell de byte

tempsLEDBat:    ds      2                        ;comptador de temps LED d'indicació de bateria correcta
tempsLED:       ds      2                        ;comptador de temps LED
tempsMot:       ds      2                        ;comptador de temps Motor
tempsRF:        ds      1                        ;comptador de temps RF
tempsBat:       ds      2                        ;comptador de temps per a lectura de bateria
resincr:       ds      1                        ;comptador de temps per resincronització
bateria:       ds      1                        ;última lectura de la bateria
misRF:         ds      8                        ;missatge rebut per RF
misResp:       ds      1                        ;missatge a tornar a l'emissor
contRF:        ds      1                        ;comptador per al missatge per RF
nByte:         ds      1                        ;byte del missatge on es troba l'estat del rx
nBit:          ds      1                        ;bit dins del byte on es troba l'estat del rx
tipusMis:      ds      1                        ;indica el tipus de missatge rebut per RF
;00-estat,011-fixar nou codi,001-petició identif
;101-petició d'estat i nivell de bateria
;111-canal de retorn d'error cap a la unitat central

```



```

data          at      0x20          ;espai de memòria adreçable a nivell de byte

flags:        ds      2            ;flags del node actuator
tramaRF       bit     flags.0      ;indica si s'ha rebut una nova trama. 1-rebuda, 0-no rebuda
consum        bit     flags.1      ;mode de baix consum usat : 1-idle, 0-moderat
estatVib      bit     flags.2      ;estatVib=0 cap motor en mode no nul. En cas contrari,estatVib=1
LEDBat        bit     flags.3      ;LEDBat-1 últim cop led en off.LEDBat-0 últim com led en on
LEDEst        bit     flags.4      ;LEDEst-1 últim cop led en off.LEDEst-0 últim cop led en on
LEDNoEst      bit     flags.5      ;LEDNoEst-1 últim cop led en off.LEDNoEst-0 últim cop led en on
flagCom       bit     flags.6      ;flagCom=1 si s'està processant mis de nou codi o lectura de codi o estat
flagLectEst   bit     flags.7      ;flagLectEst=1 si s'està processant un mis de lect d'estat/bateria
minCons       bit     flags+1.0    ;minCons=1 si es desitja minimitzar el consum
;flagLBT      bit     flags+1.1    ;flagLBT=1 si el CD=1 (si es detecta que el canal està sent usat)

bdata          ;espai de memòria adreçable a nivell de bit
node:         ds      1            ;indica l'identificador del node (bits 6:2)
;            ;i el seu estat(bits 1:0),mode de vibració,bit 7=1 si V<2.IV,0 si
V>=2.IV
idata         at      0x80

stack:        ;stack

code at 0      ;vectors d'interruptio (inici)
               ljmp    inici
code at 0x63   ;vectors d'interruptio (RTC)
               ljmp    siRTC
code at 0x53   ;vectors d'interruptio (tranceiver RF)
               ljmp    siRF
code           ;codi reubicable

;programa principal (RTC inicialment aturat i a 0)

inici:        mov     SP,#stack-1   ;inicia stack pointer
               mov     SPICLK,#0x05 ;SPICLK configurat a CPU_CLK/32
               mov     SPI_CTRL,#0x01 ;SPI connectat al port P1 (boot)
               clr     EECSN        ;comença comanda en bus SPI
               mov     a,#0x03      ;a=comanda read
               acall    spi_wr       ;tramet comanda
               mov     a,#0xFF      ;a=MSB de l'adreça
               acall    spi_wr       ;tramet byte
               mov     a,#0xFF      ;a=LSB de l'adreça
               acall    spi_wr       ;tramet byte
               acall    spi_rd       ;a=identificador del node=000A BCDE
               setb     EECSN        ;fi comanda en bus SPI
               mov     DPTR,#0xFFEh ;DPTR=0x0FFEh

```



```

movx    @DPTR,a
rl       a
rl       a
anl      a,#0x7C
jnz      noNul          ;comprovem si el codi emmagatzemat val 0
mov      a,#0x04        ;si el codi fos 0(inexistent)li forcem un 1
noNul:   mov      node,a    ;node=0ABC DE00
mov      SPICLK,#0x00    ;SPICLK configurat a CPU_CLK/2
mov      SPI_CTRL,#0x02  ;SPI connectat al transceiver RF
clr      SBSCSN          ;comença comanda en bus SPI
mov      a,#0x09        ;comanda escriptura a partir de registre de configuracio 9 (CRC &
CLK)
acall    spi_wr          ;tramet comanda
mov      a,#0xdf         ;a=dada per registre 9 (CRC & CLK)
acall    spi_wr          ;tramet dada (2 bytes CRC, cristall=16 MHz, CPU_CLK=cristall)
setb     SBSCSN          ;fi comanda en bus SPI
mov      CKLFCN,#0x1f    ;cristall=16 MHz i CPU_CLK=cristall, queda garantit CKLF de 4 kHz
mov      TICK_DV,#0x01   ;queda garantit TICK de 0,5 ms
mov      P0_ALT,#0x00    ;P0 configurat com E/S generals
mov      P0_DIR,#0x00    ;P0: bits parells configurats com sortides (lògica negativa)
                        ;P0: bits senars configurats com sortides (lògica positiva)
mov      P0_DRV,#0xff    ;P0: bits parells absorbeixen corrents elevats
                        ;P0: bits senars donen corrents elevats
mov      P0,#0x50        ;apaga tots els led
acall    initRF          ;crida a la inicialització comunicació RF
clr      TX_EN           ;selecciona transceiver en recepció
setb     TRX_CE          ;habilita transceiver
llac0:   mov      a,REGX_CTRL ;a=REGX_CTRL
jnb      ACC.4,llac0     ;espera fins que REGX_CTRL.4 sigui 0
mov      REGX_MSB,#0
mov      REGX_LSB,#20-1
mov      REGX_CTRL,#0x0a;defineix periodicitat RTC a 10 ms i l'engega
                        ;nomes cal definir 16 LSB (8 MSB a 0 per defecte)
clr      tramaRF         ;inicia tramaRF a trama no rebuda
clr      estatVib        ;inicia estat vibrador a no funcionament
setb     consum          ;inicia mode baix consum a idle
clr      minCons         ;inicia a mode diferent de la configuració de consum mínim
setb     LEDBat          ;inicia últim LEDBat en on per començar en on
setb     LEDEst          ;inicia últim LEDEst en off per començar en on
setb     LEDNoEst        ;inicia últim LEDNoEst en off per començar en on
clr      flagCom          ;no s'està processant cap mis de nou codi, lectura estat/codi
clr      flagLectEst      ;no s'està processant cap mis de lectura d'estat i bateria
;clr      flagLBT         ;iniciem flagLBT a 0:cap senyal està present en el nostre canal
mov      tempsLEDBat,#0/256;inicia tempsLEDBat=990*10 ms=9,9 s
mov      tempsLEDBat+1,#0 MOD 256
mov      tempsLED,#0/256  ;inicia tempsLED=0*10 ms=0 s
mov      tempsLED+1,#0 MOD 256

```



```

mov     tempsMot,#10/256 ;inicia tempsMot=10*10 ms=100ms
mov     tempsMot+1,#10 MOD 256
mov     tempsBat,#1000/256 ;inicialització del comptador de temps de bateria
mov     tempsBat+1,#1000 MOD 256 ;a 1000*10ms=10 s
mov     tempsRF,#1 ;inicia tempsRF=1*10 ms (10ms)
mov     resincr,#0 ;inicia resincr=256 missatges
mov     contRF,#0
mov     tipusMis,#0x00
setb    EWDI ;habilita interrupcio wakeup (RTC)
setb    EX4 ;habilita interrupcio tranceptor RF
setb    EA ;habilita interrupcions
llac1:  mov     a,REGX_CTRL ;a=REGX_CTRL
        jb     ACC.4,llac1 ;espera fins que REGX_CTRL.4 sigui 0
llac2:  jb     minCons,ebc_mpd ;mode deep power down
        jb     consum,ebc_idle ;mode baix consum a idle
ebc_mpd: mov     CK_CTRL,#0x02 ;força power down moderat
        sjmp   llac2 ;pas a power down moderat
ebc_idle: mov     PCON,#0x01 ;força idle
        sjmp   llac2 ;pas a idle

```

;servei d'interrupció del RTC

```

siRTC:  clr     WDTI ;esborra interrupcio del RTC
        push   acc
        push   b
        push   psw
        jb     minCons, siDeep0 ;entrem en configuració de mínim consum que permet intermitència
        djnz   tempsRF,siRTC1 ;no fi tempsRF, no cal fer res
        jnb    consum,siRTC0 ;fi tempsRF, i el mode era baix consum moderat
        clr     tramaRF ;el mode era idle, per tant, indiquem trama no rebuda
siRTC0: setb    TRX_CE ;habilita transceiver
        setb    consum ;actualitza mode baix consum a idle
        mov     tempsRF,#1 ;inicia tempsRF=1*10 ms (10 ms)
siRTC1: acall   lectbateria ;lectura del V del supply del modul
        ;VDD=3.66*ADCDATA/2^N, on N=6 (RESCTRL=0)
        mov     c,flagCom ;si estem senyalitzant un missatge de nou codi o de lectura
        jc     siRTC2 ;d'estat o d'identificador no superposarem la intermitència
        mov     a,node
        anl     a,#0x03 ;a=0000 00GH
        jnz    siRTC2 ;si estem en estat no nul no superposarem la intermitència
        djnz   tempsLEDBat+1,siRTC2;no fi temps
        dec     tempsLEDBat ;temps(lsb)=0, decrementa temps(msb)
        mov     a,tempsLEDBat ;a=temps(msb)
        inc     a ;a=temps(msb) anterior
        jnz    siRTC2 ;no fi temps, no cal fer res mes
        mov     c,LEDBat ;c=últim estat del LEDBat

```



```

        jnc     LEDOff
        mov     c,node.7
        mov     P00,c           ;si Vsupply>=2.1, node.7=0 i P00=0,P02=1(led verd on)
        cpl     c
        mov     P02,c           ;si node.7=1 i Vsupply<2.1, fa la intermitència P00(roig)
        mov     tempsLEDBat,#10/256;leds encesos, temps=10*10 ms (0,1 s)
        mov     tempsLEDBat+1,#10 MOD 256
        clr     LEDBat
        sjmp    siRTC2
siDeep0:
        jmp     siDeep
LEDOff:
        clr     P00             ;led P00 apagat
        clr     P02             ;led P02 apagat
        mov     tempsLEDBat,#990/256 ;leds apagats, temps=990*10ms (9,9 s)
        mov     tempsLEDBat+1,#990 MOD 256
        setb    LEDBat
siRTC2:
        djnz    tempsLED+1,siRTC3 ;no fi temps, no cal fer res.
        dec     tempsLED         ;temps(lsb)=0, decrementa temps(msb)
        mov     a,tempsLED       ;a=temps(msb)
        inc     a                 ;a=temps(msb) anterior
        jnz     siRTC3           ;no fi temps, no cal fer res mes
        mov     c,flagCom
        jc      misNoEstat
        mov     a,tipusMis
        subb    a,#0x05
        jz      actVib
        mov     a,tipusMis
        jnz     misNoEstat
actVib:
        mov     a,node           ;a=ABCD EFGH (es tracta d'un missatge d'estat)
        anl     a,#0x03           ;a=0000 00GH
        jnz     estNoNul
        mov     tempsLED,#0/256   ;inicia tempsLED=0*10 ms=0 s
        mov     tempsLED+1,#0 MOD 256
        mov     c,LEDEst
        jc      siRTC3
        clr     P02               ;estat=00, la única intermitència és la d'alimentació
        clr     P00
        setb    LEDEst
siRTC3:
        acall    motor            ;actualitzem l'estat dels vibradors segons node
        sjmp    siRTC5
estNoNul:
        mov     c,LEDEst
        jnc     vibOn
        mov     c,node.7         ;si Vsupply>=2.1 node.7=0, Vsupply<2.1 node.7=1
        mov     P00,c           ;si Vsupply>=2.1, node.7=0 i P00=0,P02=1(led verd on)
        cpl     c
        mov     P02,c           ;si node.7=1 i Vsupply<2.1, fa la intermitència P00(roig)
        mov     tempsLED,#10/256 ;leds encesos, temps=10*10 ms (0,1 s)
        mov     tempsLED+1,#10 MOD 256

```



```

        clr    LEDEst
        acall  motor                ;actualitzem l'estat dels vibradors segons node
        sjmp   siRTC5
vibOn:   clr    P00                ;led P00 apagat
        clr    P02                ;led P02 apagat
        setb   LEDEst
        mov    tempsLED,#40/256    ;si estat no nul,temps leds apagats =40*10 ms (0,4 s)
        mov    tempsLED+1,#40 MOD 256
        acall  motor                ;actualitzem l'estat dels vibradors segons node
        sjmp   siRTC5
misNoEstat: mov    c,LEDNoEst
        jnc    idenOff            ;mirem si els leds estan encesos o no
        setb   P00
        setb   P02                ;color carbassa (led vermell i verd en on)
        clr    LEDNoEst
        mov    tempsLED,#10/256    ;leds encesos, temps=10*10 ms (0,1 s)
        mov    tempsLED+1,#10 MOD 256
        mov    c,flagLectEst
        jnc    siRTC5
        acall  motor                ;actualitzem l'estat dels vibradors segons node
        sjmp   siRTC5
idenOff: clr    P00                ;led P00 apagat
        clr    P02                ;led P02 apagat
        setb   LEDNoEst            ;si és missatge de lectura d'estat i bateria realitzem la
        mov    tempsLED,#990/256   ;leds apagats, temps=990*10ms (9,9 s)
        mov    tempsLED+1,#990 MOD 256
        mov    c,flagLectEst
        jnc    siRTC4
        clr    flagLectEst
        clr    flagCom
        acall  motor                ;actualitzem l'estat dels vibradors segons node
        sjmp   siRTC5
siRTC4:  setb   minCons            ;intermitència en carbassa un cop i continuem amb normalitat
        mov    CK_CTRL,#0x02       ;pas a deep power down
siRTC5:  pop     psw
        pop     b
        pop     acc
        reti
siDeep:  djnz   tempsLED+1,fiDeep   ;no fi temps, no cal fer res.
        dec    tempsLED            ;temps(lsb)=0, decrementa temps(msb)
        mov    a,tempsLED          ;a=temps(msb)
        inc    a                   ;a=temps(msb) anterior
        jnz    fiDeep              ;no fi temps, no cal fer res mes
        mov    c,P00
        jc     araOff
        setb   P00

```




```

        setb    P02
        mov     tempsLED,#10/256          ;leds encesos, temps=10*10 ms (0,1 s)
        mov     tempsLED+1,#10 MOD 256
        sjmp    fiDeep
araOff:  clr     P00
        clr     P02
        mov     tempsLED,#990/256         ;leds apagats, temps=990*10ms (9,9 s)
        mov     tempsLED+1,#990 MOD 256
fiDeep:  pop     psw                      ;si minCons=1 es realitza únicament una intermitència,
        pop     b                        ;forçant el mode de menor consum que permet efectuar-la, el
        pop     acc                      ;qual és el mode moderate power down, i no s'efectua cap altra
        reti                                ;operació

;servei d'interrupció del transceiver de RF

siRF:    push    acc
        push    b
        push    psw
        mov     a,EXIF                  ;a=flags interrupcions esteses
        clr     ACC.6                   ;esborra interrupcio transceiver de RF
        mov     EXIF,a                  ;actualitza flags interrupcions esteses
        clr     SBCSN                   ;comença comanda en bus SPI
        mov     a,#0x24                 ;comanda de lectura de dades d'un missatge(nombre de dades a
RX_PW)
        acall    spi_wr                  ;tramet comanda
llac3:   mov     a,#8
        subb    a,contRF
        jz      misRebut
        mov     a,contRF
        add     a,#misRF
        mov     r0,a                    ;r0=#misRF+contRF
        acall    spi_rd
        mov     @r0,a
        inc     contRF
        sjmp    llac3
misRebut: setb    SBCSN                  ;fi comanda en bus SPI
        clr     TRX_CE                  ;deshabilita transceiver
        mov     contRF,#0
        mov     a,misRF
        mov     c,acc.0
        jnc     estat                   ;actualitza l'estat a la variable node
        anl     a,#0x07                  ;a=0000 0FGH
        mov     tipusMis,a              ;tipusMis=0000 0FGH
        dec     a
        jz      infoIdent               ;si tipusMis=001 és missatge de petició d'identificador
        dec     a
        dec     a

```



```

        jz      nouCodi1      ;si tipusMis=011 és missatge de nou codi
        dec     a
        dec     a
estat:   jz      infoEstat     ;si tipusMis=101 és missatge de petició d'estat i bateria
        mov     tipusMis,#0x00
        mov     a,node
        anl     a,#0x70      ;a=0ABC 0000
        swap    a            ;a=0000 0ABC
        mov     nByte,a      ;nByte=0000 0ABC
        add     a,#misRF
        mov     r0,a         ;r0=nByte+#misRF
        mov     a,node
        anl     a,#0x0C      ;a=0000 DE00
        rr      a
        rr      a           ;a=0000 00DE
        mov     nBit,a       ;nBit=0000 00DE
        jz      nBit0
        dec     a
        jz      nBit1
        dec     a
        jz      nBit2
        dec     a
        jz      nBit3
nBit0:   mov     a,@r0
        mov     c,acc.0
        mov     node.0,c
        mov     c,acc.1
        mov     node.1,c
        jmp     resin
nBit1:   mov     a,@r0
        mov     c,acc.2
        mov     node.0,c
        mov     c,acc.3
        mov     node.1,c
        jmp     resin
nBit2:   mov     a,@r0
        mov     c,acc.4
        mov     node.0,c
        mov     c,acc.5
        mov     node.1,c
        jmp     resin
nBit3:   mov     a,@r0
        mov     c,acc.6
        mov     node.0,c
        mov     c,acc.7
        mov     node.1,c

```



```

presin:      jmp      resin
nouCodi1:    sjmp     nouCodi
infoIdent:    setb    flagCom
              clr     P01
              mov     a,node           ;a=ABCD EFGH,tots els vibradors parats excepte ell
              anl     a,#0x7C         ;a=0BCD EF00
              rl      a              ;a=BCDE F000
              orl     a,#0x01         ;a=BCDE F101
              mov     misResp,a       ;misResp=BCDE F101
              jmp     envResp
infoEstat:    clr     flagCom           ;el vibrador manté l'últim estat
              clr     flagLectEst
              mov     a,misRF
              anl     a,#0xF8
              mov     b,a
              mov     a,node
              rl      a
              anl     a,#0xF8
              cjne    a,b,presin       ;si no es demana l'estat d'aquest node no retornem res
              setb    flagCom
              setb    flagLectEst
              mov     a,node           ;a=ABCD EFGH
              anl     a,#0x03         ;a=0000 00GH
              rr      a              ;a=H000 000G
              rr      a              ;a=GH00 0000
              orl     a,bateria        ;a=GHA'B' C'D'E'F', (bateria està alineada a la dreta)
              mov     misResp,a       ;tots els nodes en on si volem.Es manté l'estat de tot.
              jmp     envResp
nouCodi:      setb    flagCom           ;tots els vibradors parats excepte ell
              clr     P01
              mov     a,misRF         ;a=ABCD E011 Tots els vibradors parats menys ell.
              anl     a,#0xF8         ;a=ABCD E000
              rr      a              ;a=0ABC DE00
              mov     node,a          ;node=0ABC DE00
              rl      a              ;a=ABCD E000
              orl     a,#0x03         ;a=ABCD E011
              mov     misResp,a       ;misResp=ABCDE011(nou codi, fem eco amb mateix format)
              mov     DPTR,#0FFEh     ;DPTR=0x0FFEh
              mov     a,node
              rr      a
              rr      a
              movx    @DPTR,a         ;guardem el codi a la ERAM (memoria de codi)
              mov     SPICLK,#0x05    ;SPI_CLK=CPU_CLK/32
              mov     SPI_CTRL,#0x01  ;SPI connectat al port P1 (boot)
              clr     EWDI
              clr     EA              ;deshabilitem totes les interrupcions
wait1:        clr     EECSN           ;comença comanda en bus SPI

```



```

mov    a,#0x05      ;a=comanda read status register RDSR
acall  spi_wr        ;tramet comanda
acall  spi_rd        ;a=status register=WPENXXXBP1BP0WELWIP (de major a menor pes)
setb   EECSN         ;fi comanda en bus SPI
jb     acc.0,wait1   ;WIP=1=escriptura en procés.Esperem fins WIP(write in process)=0
clr    EECSN         ;comença comanda en bus SPI
mov    a,#0x06       ;a=comanda WREN write enable
acall  spi_wr        ;tramet comanda
setb   EECSN         ;fi comanda en bus SPI
clr    EECSN         ;comença comanda en bus SPI
mov    a,#0x02       ;a=comanda write
acall  spi_wr        ;tramet comanda
mov    a,#0xFF       ;a=MSB de l'adreça
acall  spi_wr        ;tramet byte
mov    a,#0xFF       ;a=LSB de l'adreça (a 0 hi ha un ljmp inici)
acall  spi_wr        ;tramet byte
mov    a,node        ;a=0ABC DE00 byte que conté l'identificador (1 byte de data)
rr     a
rr     a
acall  spi_wr        ;tramet identificador (data a emmagatzemar en l'EEPROM)
setb   EECSN         ;fi comanda en bus SPI
wait2: clr    EECSN         ;comença comanda en bus SPI
mov    a,#0x05       ;a=comanda read status register RDSR
acall  spi_wr        ;tramet comanda
acall  spi_rd        ;a=status register=WPENXXXBP1BP0WELWIP (de major a menor pes)
setb   EECSN         ;fi comanda en bus SPI
jb     acc.0,wait2   ;WIP=1=escriptura en procés. Esperem fins WIP(write in process)=0
clr    EECSN         ;comença comanda en bus SPI
mov    a,#0x03       ;a=comanda read
acall  spi_wr        ;tramet comanda
mov    a,#0xFF       ;a=MSB de l'adreça
acall  spi_wr        ;tramet byte
mov    a,#0xFF       ;a=LSB de l'adreça
acall  spi_wr        ;tramet byte
acall  spi_rd        ;a=node
setb   EECSN         ;fi comanda en bus SPI
setb   EWDI
setb   EA            ;habilitem totes les interrupcions
mov    SPICLK,#0x00  ;SPICLK configurat a CPU_CLK/2
mov    SPI_CTRL,#0x02 ;SPI connectat al transceiver RF
rl     a
rl     a
cjne   a,node,codiError ;si no s'ha memoritzat bé tornem error 0xDF
sjmp   envResp
codiError: mov    misResp,#0xDF ;envia error en l'emmagatzematge del nou codi
envResp:  ;setb   TRX_CE        ;habilita el transceiver, i es troba en recepció

```



;si volguéssim aplicar un protocol LBT per a impedir col·lisions de les respostes dels actuadors

```

;envResp1:      mov    c,CD           ;implementació d'un protocol LBT
;
;               mov    flagLBT,c
;
;               jc     envResp1       ;espera que CD=0 (quan el canal estigui buit)
;               setb   TX_EN          ;selecciona el transceiver en transmissió
;               clr    SBCCSN         ;comença comanda en bus SPI
;               mov    a,#0x20        ;comanda escriptura de les dades d'un missatge, escriu payload d'1 byte
;               acall   spi_wr         ;tramet comanda
;               mov    a,misResp       ;hi ha payload com a emissor d'un byte
;               acall   spi_wr         ;tramet dada 000A BCDE
;               setb   SBCCSN         ;fi comanda en bus SPI
;               setb   TRX_CE         ;habilita transceiver
envResp1:       jb     DR,envResp1    ;espera transmissió del preàmbul
envResp2:       jnb    DR,envResp2    ;espera transmissió del paquet (1 byte)
;               clr    TRX_CE         ;deshabilita transceiver
;               clr    TX_EN          ;selecciona el transceiver en recepció
resin:          clr    consum         ;actualitza mode baix consum a moderat
;               jnb    tramaRF,resin1 ;trama anterior no rebuda, cal resincronitzar
;               mov    tempsRF,#10    ;trama anterior rebuda. Inicia tempsRF=10*10 ms (100 ms)
;               djnz   resincr,resin5 ;no fi temps resincronitzacio,encara no cal resincronitzar
resin1:         mov    REGX_CTRL,#0x0 ;atura RTC (cal resincronitzar per trama no rebuda)
;               setb   tramaRF         ;inicia trama anterior rebuda
;               mov    tempsRF,#9      ;resincronització per temps. tempsRF=9*10ms(90 ms + 10ms perduts)
resin2:         mov    a,REGX_CTRL    ;fi temps resincronitzacio,a=REGX_CTRL
;               jb     ACC.4,resin2    ;espera fins que REGX_CTRL.4 sigui 0
;               mov    b,#14          ;espera uns 0,25*B=3,5ms
;               clr    a
resin3:         djnz   ACC,resin3
;               djnz   B,resin3
;               mov    REGX_MSB,#0
;               mov    REGX_LSB,#20-1
;               mov    REGX_CTRL,#0x0a;defineix periodicitat RTC a 10ms i l'engega
;                               ;nomes cal definir 16 LSB (8 MSB a 0 per defecte)
resin4:         mov    a,REGX_CTRL    ;a=REGX_CTRL
;               jb     acc.4,resin4    ;espera fins que REGX_CTRL.4=0
resin5:         pop    psw
;               pop    b
;               pop    acc
;               reti

```

;subrutina motor

;segons la variable node actualitza el mode de funcionament del mòdul

```

;recursos:      a,psw.
;
;               1 nivell de stack

```

```

motor:          mov    a,node
;               anl    a,#0x03        ;a=0000 00GH

```



```

jz      mode0
dec     a
jz      mode1on
dec     a
jz      mode2on
dec     a
jz      mode3on
mode0:  clr     estatVib
        clr     P01
        ret
mode1on: djnz    tempsMot+1,fiMotor;no fi temps, no cal fer res
        dec     tempsMot      ;temps(lsb)=0, decrementa temps(msb)
        mov     a,tempsMot     ;a=temps(msb)
        inc     a              ;a=temps(msb) anterior
        jnz     fiMotor        ;no fi temps, no cal fer res mes
        jb      P01,mode1off    ;motor funcionant. Ara el parem
        mov     tempsMot,#10/256 ;inicia temps en on=10*10ms=100ms
        mov     tempsMot+1,#10 MOD 256
        setb    P01            ;motor parat. Ara l'encenem
        setb    estatVib
        ret
mode1off: clr     P01
        mov     tempsMot,#40/256 ;inicia temps en off=40*10 ms=400ms
        mov     tempsMot+1,#40 MOD 256
        ret
mode2on: djnz    tempsMot+1,fiMotor;no fi temps, no cal fer res
        dec     tempsMot      ;temps(lsb)=0, decrementa temps(msb)
        mov     a,tempsMot     ;a=temps(msb)
        inc     a              ;a=temps(msb) anterior
        jnz     fiMotor        ;no fi temps, no cal fer res mes
        jb      P01,mode2off    ;motor funcionant. Ara el parem
        mov     tempsMot,#10/256 ;inicia temps en on=10*10ms=100ms
        mov     tempsMot+1,#10 MOD 256
        setb    P01            ;motor parat. Ara l'encenem
        setb    estatVib
        ret
mode2off: clr     P01
        mov     tempsMot,#10/256 ;inicia temps en off=10*10 ms=100ms
        mov     tempsMot+1,#10 MOD 256
        ret
mode3on: djnz    tempsMot+1,fiMotor;no fi temps, no cal fer res
        dec     tempsMot      ;temps(lsb)=0, decrementa temps(msb)
        mov     a,tempsMot     ;a=temps(msb)
        inc     a              ;a=temps(msb) anterior
        jnz     fiMotor        ;no fi temps, no cal fer res mes
        jb      P01,mode3off    ;motor funcionant. Ara el parem

```



```

mov     tempsMot,#2/256    ;inicia temps en on=2*10ms=20ms
mov     tempsMot+1,#2 MOD 256
setb    P01                ;motor parat. Ara l'encenem
setb    estatVib
ret
mode3off:  clr    P01
mov     tempsMot,#2/256    ;inicia temps en off=2*10 ms=20ms
mov     tempsMot+1,#2 MOD 256
fiMotor:  ret

```

;subrutina initRF

;inicialitza els paràmetres relatius a les transmissions per RF

```

;recursos:      psw
;               1 nivell de stack

```

```

initRF:  clr    SBCSN        ;comença comanda en bus SPI
mov     a,#0x00             ;comanda escriptura a partir de registre config 0 (CH_NO)
acall   spi_wr              ;tramet comanda
mov     a,#0x7C              ;a=dada per registre 0 (CH_NO)
acall   spi_wr              ;tramet dada (canal 124 => 869,6 MHz)
mov     a,#0x0e              ;a=dada per registre 1 (varis)
acall   spi_wr              ;tramet dada (potencia transmissio maxima,
                           ;no retransmissio, recepcio normal i 868/915 MHz)
mov     a,#0x44              ;a=dada per registre 2 (TX_RX_AFW)
acall   spi_wr              ;tramet dada (adreces de 4 bytes)
mov     a,#0x08              ;a=dada per registre 3 (RX_PW)
acall   spi_wr              ;tramet dada (RX payload de 8 bytes)
mov     a,#0x01              ;a=dada per registre 4 (TX_PW)
acall   spi_wr              ;tramet dada (TX payload de 1 byte)
mov     a,#0xA5              ;a=dada per registre 5 (RX_ADDRESS byte 0)
acall   spi_wr              ;tramet dada
mov     a,#0xED              ;a=dada per registre 6 (RX_ADDRESS byte 1)
acall   spi_wr              ;tramet dada
mov     a,#0xE6              ;a=dada per registre 7 (RX_ADDRESS byte 2)
acall   spi_wr              ;tramet dada
mov     a,#0xDE              ;a=dada per registre 8 (RX_ADDRESS byte 3)
acall   spi_wr              ;tramet dada
setb    SBCSN              ;fi comanda en bus SPI
clr     SBCSN
mov     a,#0x22              ;a=comanda d'escriptura de TX_ADDRESS des de byte 0
acall   spi_wr              ;tramet comanda
mov     a,#0xA5              ;a=dada per a byte 0 (TX_ADDRESS byte 0)
acall   spi_wr              ;tramet dada
mov     a,#0xED              ;a=dada per a byte 1 (TX_ADDRESS byte 1)
acall   spi_wr              ;tramet dada
mov     a,#0xE6              ;a=dada per a byte 2 (TX_ADDRESS byte 2)
acall   spi_wr              ;tramet dada

```



```

    mov    a,#0xDE      ;a=dada per a byte 3 (TX_ADDRESS byte 3)
    acall  spi_wr        ;tramet dada
    setb   SBCSN        ;fi comanda en bus SPI
    ret

;subrutina inibateria
;inicialitza el registre ADC-Configuration a fi de monitoritzar el voltatge
;que rep el modul nRF9E5.
;recursos:      a,psw
;                2 nivells de stack

inibateria:
    clr    SBCSN        ;comença comanda en bus SPI
    mov    a,#0x44      ;comanda d'escriptura del registre ADC configuration,el qual son 2
    bytes

    acall  spi_wr        ;tramet comanda
    mov    a,#0x85      ;a=monitoritzacio de VDD/3, PWR_UP=1,VFSEL=0
    acall  spi_wr        ;tramet registre de configuracio
    mov    a,#0x08      ;a=ADC_RL_JUST=1 justificacio a dreta, DIFFMODE=0,
                        ;RESCTRL=0->6 bits. Aixi, ADCDATA[5:0]

    acall  spi_wr        ;tramet registre de configuracio
    setb   SBCSN        ;fi comanda en bus SPI
    ret

;subrutina lectbateria
;llegeix el V supply (analogic) i guarda a l'acumulador VDD/3 en format digital un cop cada 10 segons
;recursos:      a,psw.
;                2 nivells de stack

lectbateria:
    djnz   tempsBat+1,fiLect
    dec    tempsBat
    mov    a,tempsBat
    inc    a
    jnz    fiLect
    acall  inibateria
    clr    SBCSN        ;comença comanda en bus SPI
    mov    a,#0xc8      ;a=comanda d'inici de conversio ADC de l'entrada VDD/3
    acall  spi_wr        ;tramet comanda
    setb   SBCSN        ;fi comanda en bus SPI
fiADC:
    jnb    EOC,fiADC    ;espera fins que la conversio sigui completa
    clr    SBCSN        ;comença comanda en bus SPI
    anl    EXIF,#0xf7   ;esborra interrupcio ADC: IE2=0
    mov    a,#0x40      ;a=comanda de lectura ADC data,començant pel byte 0
    acall  spi_wr        ;tramet comanda
    acall  spi_rd        ;a=ADC DATA byte 0 (byte on es guarda la conversio)
    setb   SBCSN        ;fi comanda en bus SPI
    mov    bateria,a
    clr    SBCSN        ;comença comanda en bus SPI

```




```

bytes
    mov    a,#0x44      ;comanda d'escriptura del registre ADC configuration,el qual son 2
    acall  spi_wr        ;tramet comanda
    mov    a,#0x81      ;a=monitoritzacio de VDD/3, PWR_UP=0,VFSEL=0
    acall  spi_wr        ;tramet comanda
    mov    a,#0x09
    acall  spi_wr
    setb   SBCSN        ;fi comanda en bus SPI
    clr    c
    mov    a,bateria
    subb   a,#37d        ;resta 2.1V(aprox)de a=Vsupply amb bit de portar-ne
    mov    node.7,c      ;si Vsupply>=2.1 node.7=0, Vsupply<2.1 node.7=1
    mov    tempsBat,#1000/256;inicialització del comptador de temps de bateria
    mov    tempsBat+1,#1000 MOD 256 ;a 1000*10ms=10 s
fiLect:    ret

;subrutina spi_rd
;rep un byte del bus SPI.    par. sortida:    a=byte rebut
;recursos:    psw
;                1 nivell de stack

;subrutina spi_wr
;envia un byte pel bus SPI    par. entrada:    a=byte a enviar
;recursos:    a,psw
;                1 nivell de stack

spi_rd:
spi_wr:    anl    EXIF,#0xdf        ;esborra interrupcio SPI
    mov    SPI_DATA,a        ;SPI_DATA=dada a enviar (no caldria a spi_rd)
spi_rdwr1:    mov    a,EXIF        ;llegeix SPI_RX (EXIF.5)
    jnb    ACC.5,spi_rdwr1    ;espera fins que SPI_RX=1
    mov    a,SPI_DATA        ;a=SPI_DATA (no caldria a spi_wr)
    ret
end

```





K. Codi implementat en els nodes de 3 vibradors

```

;= nodes actuadors de tres vibradors.a51 =====
;-cada 100 ms mira si ha rebut un missatge de RF (usa la interrupció 0x53 de RF)
;-resincronització cada 256 trames
;-llegeix l'estat de la bateria i el guarda a node.7(node.7=0 si Vsupply>=2.1,node.7=1 si V<2.1)
;LEDS:
;-si tots els vibradors parats:intermitència 0.1s on, 9.9 off a P02 si Vsupply és alt (en verd)
;-i a P00 si Vsupply és baix (en roig), és a dir, segons node.7.
;-si algun vibrador en on: el led estarà 100ms encés,400ms parat segons Vsupply(node.7)
;-si comanda nova identificació completada:P00 i P02 0.1s en on a la vegada, 9.9 off
;-si comanda petició identificació:el led s'il·luminarà 500ms segons Vsupply (node.7)
;-si comanda petició estat i bateria:el led s'il·luminarà 500ms segons Vsupply (node.7)
;MOTORS:
;-mode 00: repòs. Llavors estatVib = 0 (tots els motors parats)
;-mode 01: 100ms vibrant cada 500 ms (f=2Hz, D=20%). estatVib = 1 (algun motor en mode no nul)
;-mode 10: 100ms vibrant cada 200 ms (f=5Hz, D=50%). estatVib = 1 (algun motor en mode no nul)
;-mode 11: 20ms vibrant cada 40 ms (f=25Hz, D=50%). estatVib = 1 (algun motor en mode no nul)
;=====

```

\$nomod51

\$include(nRF9E5.inc)

;assignacions dels SFR del nRF9E5

data

;espai de memòria adreçable a nivell de byte

tempsLEDBat:	ds	2	;comptador de temps LED d'indicació de bateria correcta
tempsLED:	ds	2	;comptador de temps LED
tempsMot:	ds	6	;comptador de temps Motor
tempsRF:	ds	1	;comptador de temps RF
tempsBat:	ds	2	;comptador de temps per a lectura de bateria
resincr:	ds	1	;comptador de temps per resincronització
bateria:	ds	1	;última lectura de la bateria
misRF:	ds	8	;missatge rebut per RF
misResp:	ds	1	;missatge a tornar a l'emissor
contRF:	ds	1	;comptador per al missatge per RF
contNode:	ds	1	;comptador per al número de node
nByte:	ds	1	;byte del missatge on es troba l'estat del rx
nBit:	ds	1	;bit dins del byte on es troba l'estat del rx
tipusMis:	ds	1	;indica el tipus de missatge rebut per RF

;00-estat,011-fixar nou codi,001-petició identif

;101-petició d'estat i nivell de bateria

;111-canal de retorn d'error cap a la unitat central

data at 0x20

;espai de memòria adreçable a nivell de byte



flags:	ds	2	<i>;flags del node actuator</i>
tramaRF	bit	flags.0	<i>;indica si s'ha rebut una nova trama. 1-rebuda, 0-no rebuda</i>
consum	bit	flags.1	<i>;mode de baix consum usat : 1-idle, 0-moderat</i>
estatVib	bit	flags.2	<i>;estatVib=0 cap motor en mode no nul. En cas contrari,estatVib=1</i>
LEDBat	bit	flags.3	<i>;LEDBat-1 últim cop led en off.LEDBat-0 últim com led en on</i>
LEDEst	bit	flags.4	<i>;LEDEst-1 últim cop led en off.LEDEst-0 últim cop led en on</i>
LEDNoEst	bit	flags.5	<i>;LEDNoEst-1 últim cop led en off.LEDNoEst-0 últim cop led en on</i>
flagCom	bit	flags.6	<i>;flagCom=1 si s'està processant mis de nou codi o lectura de codi o estat</i>
flagLectEst	bit	flags.7	<i>;flagLectEst=1 si s'està processant un mis de lect d'estat/bateria</i>
minCons	bit	flags+1.0	<i>;minCons=1 si es desitja minimitzar el consum</i>
<i>;flagLBT</i>	bit	<i>flags+1.1</i>	<i>;flagLBT=1 si el CD=1 (si es detecta que el canal està sent usat)</i>

bdata *;espai de memòria adreçable a nivell de bit*

node: **ds** 3 *;indica l'identificador dels nodes (bits 6:2 de cada byte); el seu estat*
;(bits 1:0 de cada byte).El bit 7 del byte 0=1 si V<2.1V,0 si V>=2.1V

idata at 0x80

stack: *;stack*

code at 0 *;vectors d'interruptio (inici)*

ljmp inici *;ljmp ocupa 3 bytes*

code at 0x63 *;vectors d'interruptio (RTC)*

ljmp siRTC

code at 0x53 *;vectors d'interruptio (tranceiver RF)*

ljmp siRF

code *;codi reubicable*

;programa principal (RTC inicialment aturat i a 0)

inici:	mov	SP,#stack-1	<i>;inicia stack pointer</i>
	mov	SPICLK,#0x05	<i>;SPICLK configurat a CPU_CLK/32</i>
	mov	SPI_CTRL,#0x01	<i>;SPI connectat al port P1 (boot)</i>
	clr	EECSN	<i>;comença comanda en bus SPI</i>
	mov	a,#0x03	<i>;a=comanda read</i>
	acall	spi_wr	<i>;tramet comanda</i>
	mov	a,#0xFF	<i>;a=MSB de l'adreça</i>
	acall	spi_wr	<i>;tramet byte</i>
	mov	a,#0xFF	<i>;a=LSB de l'adreça</i>
	acall	spi_wr	<i>;tramet byte</i>
	acall	spi_rd	<i>;a=identificador del node=000A BCDE</i>
	setb	EECSN	<i>;fi comanda en bus SPI</i>
	mov	DPTR,#0xFFEh	<i>;DPTR=0xFFEh</i>
	movx	@DPTR,a	
	rl	a	
	rl	a	



```

noNul:    anl      a,#0x7C
          jnz      noNul      ;comprovem si el codi emmagatzemat val 0
          mov      a,#0x04     ;si el codi fos 0(inexistent)li forcem un 1
          mov      node,a      ;node=0ABC DE00
          add      a,#0x04
          mov      node+1,a
          add      a,#0x04
          mov      node+2,a
          mov      SPICLK,#0x00 ;SPICLK configurat a CPU_CLK/2
          mov      SPI_CTRL,#0x02 ;SPI connectat al transceiver RF
          clr      SBCCSN      ;comença comanda en bus SPI
          mov      a,#0x09     ;comanda d'escriptura a partir de registre de configuracio 9 (CRC &
CLK)
          acall     spi_wr      ;tramet comanda
          mov      a,#0xdf     ;a=dada per registre 9 (CRC & CLK)
          acall     spi_wr      ;tramet dada (2 bytes CRC, cristall=16 MHz, CPU_CLK=cristall)
          setb      SBCCSN      ;fi comanda en bus SPI
          mov      CKLFCN,#0x1f ;cristall=16 MHz i CPU_CLK=cristall, queda garantit CKLF de 4 kHz
          mov      TICK_DV,#0x01 ;queda garantit TICK de 0,5 ms
          mov      P0_ALT,#0x00 ;P0 configurat com E/S generals
          mov      P0_DIR,#0x00 ;P0: bits parells configurats com sortides (lògica negativa)
          ;P0: bits senars configurats com sortides (lògica positiva)
          mov      P0_DRV,#0xff ;P0: bits parells absorbeixen corrents elevats
          ;P0: bits senars donen corrents elevats
          mov      P0,#0x50     ;apaga tots els led
          acall     initRF      ;crida a la inicialització comunicació RF
          clr      TX_EN       ;selecciona transceiver en recepció
          setb      TRX_CE      ;habilita transceiver
llac0:    mov      a,REGX_CTRL  ;a=REGX_CTRL
          jb       ACC.4,llac0  ;espera fins que REGX_CTRL.4 sigui 0
          mov      REGX_MSB,#0
          mov      REGX_LSB,#20-1
          mov      REGX_CTRL,#0x0a;defineix periodicitat RTC a 10 ms i l'enega
          ;nomes cal definir 16 LSB (8 MSB a 0 per defecte)
          clr      tramaRF      ;inicia tramaRF a trama no rebuda
          clr      estatVib     ;inicia estat vibrador a no funcionament
          setb      consum      ;inicia mode baix consum a idle
          clr      minCons      ;inicia a mode diferent de la configuració de consum mínim
          setb      LEDBat      ;inicia últim LEDBat en on per començar en on
          setb      LEDEst      ;inicia últim LEDEst en off per començar en on
          setb      LEDNoEst     ;inicia últim LEDNoEst en off per començar en on
          clr      flagCom      ;no s'està processant cap mis de nou codi, lectura estat/codi
          clr      flagLectEst   ;no s'està processant cap mis de lectura d'estat i bateria
          ;clr      flagLBT      ;iniciem flagLBT a 0:cap senyal està present en el nostre canal
          mov      tempsLEDBat,#0/256;inicia tempsLEDBat=990*10 ms=9,9 s
          mov      tempsLEDBat+1,#0 MOD 256
          mov      tempsLED,#0/256 ;inicia tempsLED=0*10 ms=0 s

```



```

mov     tempsLED+1,#0 MOD 256
mov     tempsMot,#10/256 ;inicia tempsMot=10*10 ms=100ms (vibrador 1)
mov     tempsMot+1,#10 MOD 256
mov     tempsMot+2,#10/256 ;inicia tempsMot=10*10 ms=100ms (vibrador 2)
mov     tempsMot+3,#10 MOD 256
mov     tempsMot+4,#10/256 ;inicia tempsMot=10*10 ms=100ms (vibrador 3)
mov     tempsMot+5,#10 MOD 256
mov     tempsBat,#1000/256 ;inicialització del comptador de temps de bateria
mov     tempsBat+1,#1000 MOD 256 ;a 1000*10ms=10 s
mov     tempsRF,#1 ;inicia tempsRF=1*10 ms (10ms)
mov     resincr,#0 ;inicia resincr=256 missatges
mov     contRF,#0
mov     tipusMis,#0x00
setb    EWDI ;habilita interrupcio wakeup (RTC)
setb    EX4 ;habilita interrupcio tranceptor RF
setb    EA ;habilita interrupcions
llac1:   mov     a,REGX_CTRL ;a=REGX_CTRL
        jb      ACC.4,llac1 ;espera fins que REGX_CTRL.4 sigui 0
llac2:   jb      minCons,ebc_mpd ;mode deep power down
        jb      consum,ebc_idle ;mode baix consum a idle
ebc_mpd: mov     CK_CTRL,#0x02 ;força power down moderat
        sjmp    llac2 ;pas a power down moderat
ebc_idle: mov     PCON,#0x01 ;força idle
        sjmp    llac2 ;pas a idle

;servei d'interrupció del RTC

siRTC:   clr     WDTI ;esborra interrupcio del RTC
        push    acc
        push    b
        push    psw
        jb      minCons, siDeep0 ;entrem en configuració de mín cons que permet intermitència
        djnz    tempsRF,siRTC1 ;no fi tempsRF, no cal fer res
        jnb     consum,siRTC0 ;fi tempsRF, i el mode era baix consum moderat
        clr     tramaRF ;el mode era idle, per tant, indiquem trama no rebuda
siRTC0:  setb    TRX_CE ;habilita transceiver
        setb    consum ;actualitza mode baix consum a idle
        mov     tempsRF,#1 ;inicia tempsRF=1*10 ms (10 ms)
siRTC1:  acall    lectbateria ;lectura del V del supply del modul
        ;VDD=3.66*ADCDATA/2^N, on N=6 (RESCTRL=0)
        mov     c,flagCom ;si estem senyalitzant un missatge de nou codi o de lectura
        jc      siRTC2 ;d'estat o d'identificador no superposarem la intermitència
        mov     a,node
        anl     a,#0x03 ;a=0000 00GH
        jnz     siRTC2 ;si estem en estat no nul no superposarem la intermitència

```



```

mov     a,node+1
anl     a,#0x03                ;a=0000 00GH
jnz     siRTC2                 ;si estem en estat no nul no superposarem la intermitència
mov     a,node+2
anl     a,#0x03                ;a=0000 00GH
jnz     siRTC2                 ;si estem en estat no nul no superposarem la intermitència
djnz    tempsLEDBat+1,siRTC2   ;no fi temps
dec     tempsLEDBat            ;temps(lsb)=0, decrementa temps(msb)
mov     a,tempsLEDBat          ;a=temps(msb)
inc     a                     ;a=temps(msb) anterior
jnz     siRTC2                 ;no fi temps, no cal fer res mes
mov     c,LEDBat               ;c=últim estat del LEDBat
jnc     LEDOff
mov     c,node.7
mov     P00,c                  ;si Vsupply>=2.1, node.7=0 i P00=0,P02=1(led verd on)
cpl     c
mov     P02,c                  ;si node.7=1 i Vsupply<2.1, fa la intermitència P00(roig)
mov     tempsLEDBat,#10/256    ;leds encesos, temps=10*10 ms (0,1 s)
mov     tempsLEDBat+1,#10 MOD 256
clr     LEDBat
sjmp    siRTC2
siDeep0: jmp    siDeep
LEDOff:  clr     P00             ;led P00 apagat
        clr     P02             ;led P02 apagat
        mov     tempsLEDBat,#990/256 ;leds apagats, temps=990*10ms (9,9 s)
        mov     tempsLEDBat+1,#990 MOD 256
        setb    LEDBat
siRTC2:  djnz    tempsLED+1,siRTC3 ;no fi temps, no cal fer res.
        dec     tempsLED         ;temps(lsb)=0, decrementa temps(msb)
        mov     a,tempsLED       ;a=temps(msb)
        inc     a                 ;a=temps(msb) anterior
        jnz     siRTC3           ;no fi temps, no cal fer res mes
        mov     c,flagCom
        jc      misNoEstat
        mov     a,tipusMis
        subb    a,#0x05
        jz      actVib
        mov     a,tipusMis
        jnz     misNoEstat
actVib:  mov     a,node           ;a=ABCD EFGH (es tracta d'un missatge d'estat)
        anl     a,#0x03          ;a=0000 00GH
        jnz     estNoNul
        mov     a,node+1         ;a=ABCD EFGH (es tracta d'un missatge d'estat)
        anl     a,#0x03          ;a=0000 00GH
        jnz     estNoNul
        mov     a,node+2         ;a=ABCD EFGH (es tracta d'un missatge d'estat)
        anl     a,#0x03          ;a=0000 00GH

```



```

jnz      estNoNul
mov      tempsLED,#0/256          ;inicia tempsLED=0*10 ms=0 s
mov      tempsLED+1,#0 MOD 256
mov      c,LEDEst
jc       siRTC3
clr      P02                      ;estat=00, la única intermitència és la d'alimentació
clr      P00
setb     LEDEst
siRTC3:   acall    motor              ;actualitzem l'estat dels vibradors segons node
sjmp     siRTC5
estNoNul: mov      c,LEDEst
jnc      vibOn
mov      c,node.7                  ;si Vsupply>=2.1 node.7=0, Vsupply<2.1 node.7=1
mov      P00,c                    ;si Vsupply>=2.1, node.7=0 i P00=0,P02=1(led verd on)
cpl      c
mov      P02,c                    ;si node.7=1 i Vsupply<2.1, fa la intermitència P00(roig)
mov      tempsLED,#10/256         ;leds encesos, temps=10*10 ms (0,1 s)
mov      tempsLED+1,#10 MOD 256
clr      LEDEst
acall     motor                  ;actualitzem l'estat dels vibradors segons node
sjmp     siRTC5
vibOn:   clr      P00                ;led P00 apagat
clr      P02                      ;led P02 apagat
setb     LEDEst
mov      tempsLED,#40/256         ;si estat no nul,temps leds apagats =40*10 ms (0,4 s)
mov      tempsLED+1,#40 MOD 256
acall     motor                  ;actualitzem l'estat dels vibradors segons node
sjmp     siRTC5
misNoEstat: mov      c,LEDNoEst
jnc      idenOff                  ;mirem si els leds estan encesos o no
setb     P00
setb     P02                      ;color carbassa (led vermell i verd en on)
clr      LEDNoEst
mov      tempsLED,#10/256         ;leds encesos, temps=10*10 ms (0,1 s)
mov      tempsLED+1,#10 MOD 256
mov      c,flagLectEst
jnc      siRTC5
acall     motor                  ;actualitzem l'estat dels vibradors segons node
sjmp     siRTC5
idenOff: clr      P00                ;led P00 apagat
clr      P02                      ;led P02 apagat
setb     LEDNoEst                ;si és missatge de lectura d'estat i bateria realitzem la
mov      tempsLED,#990/256        ;leds apagats, temps=990*10ms (9,9 s)
mov      tempsLED+1,#990 MOD 256
mov      c,flagLectEst
jnc      siRTC4

```




```

        clr    flagLectEst
        clr    flagCom
        acall  motor                ;actualitzem l'estat dels vibradors segons node
        sjmp   siRTC5                ;intermitència en carbassa un cop i continuem amb normalitat

siRTC4:  setb   minCons                ;pas a deep power down
        mov    CK_CTRL,#0x02          ;pas a deep power down
siRTC5:  pop     psw
        pop     b
        pop     acc
        reti

siDeep:  djnz   tempsLED+1,fiDeep      ;no fi temps, no cal fer res.
        dec    tempsLED                ;temps(lsb)=0, decrementa temps(msb)
        mov    a,tempsLED              ;a=temps(msb)
        inc    a                        ;a=temps(msb) anterior
        jnz    fiDeep                  ;no fi temps, no cal fer res mes
        mov    c,P00
        jc     araOff
        setb   P00
        setb   P02
        mov    tempsLED,#10/256        ;leds encesos, temps=10*10 ms (0,1 s)
        mov    tempsLED+1,#10 MOD 256
        sjmp   fiDeep
araOff:  clr    P00
        clr    P02
        mov    tempsLED,#990/256       ;leds apagats, temps=990*10ms (9,9 s)
        mov    tempsLED+1,#990 MOD 256
fiDeep:  pop     psw                    ;si minCons=1 es realitza únicament una intermitència,
        pop     b                        ;forçant el mode de menor consum que permet efectuar-la, el
        pop     acc                      ;qual és el mode moderate power down, i no s'efectua cap

altra
        reti                            ;operació

```

;servei d'interrupció del transceiver de RF

```

siRF:    push   acc
        push   b
        push   psw
        mov    a,EXIF                  ;a=flags interrupcions esteses
        clr    ACC.6                    ;esborra interrupcio transceiver de RF
        mov    EXIF,a                  ;actualitza flags interrupcions esteses
        clr    SBCCSN                  ;comença comanda en bus SPI
        mov    a,#0x24                  ;comanda de lectura d'un mis (nombre de dades a RX_PW)
        acall  spi_wr                  ;tramet comanda

llac3:   mov    a,#8
        subb   a,contRF
        jz     misRebut

```



```

mov    a,contRF
add    a,#misRF
mov    r0,a                ;r0=#misRF+contRF
acall  spi_rd
mov    @r0,a
inc    contRF
sjmp   llac3
misRebut:
setb   SBCSN                ;fi comanda en bus SPI
clr    TRX_CE                ;deshabilita transceiver
mov    contRF,#0
mov    a,misRF
mov    c,acc.0
jnc    estat                ;actualitza l'estat a la variable node
anl    a,#0x07                ;a=0000 0FGH
mov    tipusMis,a            ;tipusMis=0000 0FGH
dec    a
jz     infoIdent1            ;si tipusMis=001 és missatge de petició d'identificador
dec    a
dec    a
jz     nouCodi1              ;si tipusMis=011 és missatge de nou codi
dec    a
dec    a
jz     infoEstat1            ;si tipusMis=101 és missatge de petició d'estat i bateria
estat:
mov    tipusMis,#0x00
mov    contNode,#0
actnod:
mov    a,contNode
add    a,#node
mov    r1,a                ;r1=#node+contNode
mov    a,@r1                ;a=@(node+contNode)
mov    b,a
anl    a,#0x70                ;a=0ABC 0000
swap   a                    ;a=0000 0ABC
mov    nByte,a                ;nByte=0000 0ABC
add    a,#misRF
mov    r0,a                ;r0=nByte+#misRF
mov    a,@r1
anl    a,#0x0C                ;a=0000 DE00
rr     a
rr     a                    ;a=0000 00DE
mov    nBit,a                ;nBit=0000 00DE
jz     nBit0
dec    a
jz     nBit1
dec    a
jz     nBit2
dec    a

```



```

                                jz      nBit3
infoIdent1:                    jmp     infoIdent
nouCodi1:                      jmp     nouCodi
infoEstat1:                    jmp     infoEstat
nBit0:                          mov     a,@r0
                                mov     c,acc.0
                                mov     b.0,c
                                mov     c,acc.1
                                mov     b.1,c
                                inc      contNode
                                mov     @r1,b
                                mov     a,contNode
                                cjne     a,#0x03,actnod
                                jmp     resin
nBit1:                          mov     a,@r0
                                mov     c,acc.2
                                mov     b.0,c
                                mov     c,acc.3
                                mov     b.1,c
                                inc      contNode
                                mov     @r1,b
                                mov     a,contNode
                                cjne     a,#0x03,actnod
                                jmp     resin
nBit2:                          mov     a,@r0
                                mov     c,acc.4
                                mov     b.0,c
                                mov     c,acc.5
                                mov     b.1,c
                                inc      contNode
                                mov     @r1,b
                                mov     a,contNode
                                cjne     a,#0x03,actnod
                                jmp     resin
nBit3:                          mov     a,@r0
                                mov     c,acc.6
                                mov     b.0,c
                                mov     c,acc.7
                                mov     b.1,c
                                inc      contNode
                                mov     @r1,b
                                mov     a,contNode
                                cjne     a,#0x03,actnod
presin:                        jmp     resin
infoIdent:                      setb    flagCom
                                clr      P01
                                mov     a,node

```

;a=ABCD EFGH,tots els vibradors parats excepte ell



```

anl    a,#0x7C           ;a=0BCD EF00
rl     a                 ;a=BCDE F000
orl    a,#0x01           ;a=BCDE F101
mov    misResp,a         ;misResp=BCDE F101
jmp    envResp
infoEstat:
clr    flagCom           ;el vibrador manté l'últim estat
clr    flagLectEst
mov    a,misRF
anl    a,#0xF8
mov    b,a
mov    a,node            ;es retorna l'estat del node[0]
rl     a
anl    a,#0xF8
cjne   a,b,node1        ;si no es demana l'estat d'aquest node no retornem res
setb   flagCom
setb   flagLectEst
mov    a,node            ;a=ABCD EFGH
anl    a,#0x03           ;a=0000 00GH
rr     a                 ;a=H000 000G
rr     a                 ;a=GH00 0000
orl    a,bateria        ;a=GHA'B' C'D'E'F', (bateria està alineada a la dreta)
mov    misResp,a        ;tots els nodes en on si volem.Es manté l'estat de tot.
jmp    envResp
node1:
mov    a,misRF
anl    a,#0xF8
mov    b,a
mov    a,node+1          ;es retorna l'estat del node[1]
rl     a
anl    a,#0xF8
cjne   a,b,node2        ;si no es demana l'estat d'aquest node no retornem res
setb   flagCom
setb   flagLectEst
mov    a,node+1          ;a=ABCD EFGH
anl    a,#0x03           ;a=0000 00GH
rr     a                 ;a=H000 000G
rr     a                 ;a=GH00 0000
orl    a,bateria        ;a=GHA'B' C'D'E'F', (bateria està alineada a la dreta)
mov    misResp,a        ;tots els nodes en on si volem.Es manté l'estat de tot.
jmp    envResp
node2:
mov    a,misRF
anl    a,#0xF8
mov    b,a
mov    a,node+2          ;es retorna l'estat del node[2]
rl     a
anl    a,#0xF8
cjne   a,b,presin       ;si no es demana l'estat d'aquest node no retornem res

```



```

nouCodi:
    setb    flagCom
    setb    flagLectEst
    mov     a,node+2          ;a=ABCD EFGH
    anl     a,#0x03           ;a=0000 00GH
    rr      a                 ;a=H000 000G
    rr      a                 ;a=GH00 0000
    orl     a,bateria         ;a=GHA'B' C'D'E'F', (bateria està alineada a la dreta)
    mov     misResp,a         ;tots els nodes en on si volem.Es manté l'estat de tot.
    jmp     envResp
    setb    flagCom           ;tots els vibradors parats excepte ell
    clr     P01
    mov     a,misRF           ;a=ABCD E011 Tots els vibradors parats menys ell.
    anl     a,#0xF8           ;a=ABCD E000
    rr      a                 ;a=0ABC DE00
    mov     node,a            ;node=0ABC DE00
    add     a,#0x04
    mov     node+1,a
    add     a,#0x04
    mov     node+2,a
    mov     a,node
    rl      a                 ;a=ABCD E000
    orl     a,#0x03           ;a=ABCD E011
    mov     misResp,a         ;misResp=ABCDE011(nou codi, fem un eco amb igual format)
    mov     DPTR,#0FFEh       ;DPTR=0x0FFEh
    mov     a,node
    rr      a
    rr      a
    movx    @DPTR,a           ;guardem el codi a la ERAM (memoria de codi)
    mov     SPICLK,#0x05       ;SPI_CLK=CPU_CLK/32
    mov     SPI_CTRL,#0x01     ;SPI connectat al port P1 (boot)
    clr     EWDI
    clr     EA                 ;deshabilitem totes les interrupcions
    clr     EECSN              ;comença comanda en bus SPI
    mov     a,#0x05            ;a=comanda read status register RDSR
    acall    spi_wr            ;tramet comanda
    acall    spi_rd            ;a=status register=WPENXXXBP1BP0WELWIP(de + a - pes)
    setb    EECSN              ;fi comanda en bus SPI
    jnb     acc.0,wait1        ;WIP=1=escript. en procés.EsperemWIP(write in process)=0
    clr     EECSN              ;comença comanda en bus SPI
    mov     a,#0x06            ;a=comanda WREN write enable
    acall    spi_wr            ;tramet comanda
    setb    EECSN              ;fi comanda en bus SPI
    clr     EECSN              ;comença comanda en bus SPI
    mov     a,#0x02            ;a=comanda write
    acall    spi_wr            ;tramet comanda
    mov     a,#0xFF            ;a=MSB de l'adreça
    acall    spi_wr            ;tramet byte

```



```

        mov     a,#0xFF                ;a=LSB de l'adreça (a 0 hi ha un ljmp inici)
        acall   spi_wr                 ;tramet byte
        mov     a,node                 ;a=0ABC DE00 byte que conté l'identificador (1 byte de data)
        rr      a
        rr      a
        acall   spi_wr                 ;tramet identificador (data a emmagatzemar en l'EEPROM)
        setb    EECSN                  ;fi comanda en bus SPI
wait2:    clr     EECSN                 ;comença comanda en bus SPI
        mov     a,#0x05                 ;a=comanda read status register RDSR
        acall   spi_wr                 ;tramet comanda
        acall   spi_rd                 ;a=status register=WPENXXXBP1BP0WELWIP(de + a -
pes)

        setb    EECSN                  ;fi comanda en bus SPI
        jnb     acc.0,wait2             ;WIP=1=escript. en procés.EsperemWIP(write in process)=0
        clr     EECSN                 ;comença comanda en bus SPI
        mov     a,#0x03                 ;a=comanda read
        acall   spi_wr                 ;tramet comanda
        mov     a,#0xFF                 ;a=MSB de l'adreça
        acall   spi_wr                 ;tramet byte
        mov     a,#0xFF                 ;a=LSB de l'adreça
        acall   spi_wr                 ;tramet byte
        acall   spi_rd                 ;a=node
        setb    EECSN                  ;fi comanda en bus SPI
        setb    EWDI
        setb    EA                     ;habilitem totes les interrupcions
        mov     SPICLK,#0x00            ;SPICLK configurat a CPU_CLK/2
        mov     SPI_CTRL,#0x02         ;SPI connectat al transceiver RF
        rl      a
        rl      a
        cjne    a,node,codiError        ;si no s'ha memoritzat bé tornem error 0xDF
        sjmp     envResp
codiError: mov     misResp,#0xDF         ;envia error en l'emmagatzematge del nou codi
envResp:    ;setb    TRX_CE              ;habilita el transceiver, i es troba en recepció
;si volguéssim aplicar un protocol LBT per a impedir col·lisions de les respostes dels actuadors
;envResp1:    mov     c,CD                ;implementació d'un protocol LBT
;
;            mov     flagLBT,c
;            jc      envResp1            ;espera que CD=0 (quan el canal estigui buit)
        setb    TX_EN                  ;selecciona el transceiver en transmissió
        clr     SBCSN                  ;comença comanda en bus SPI
        mov     a,#0x20                 ;comanda escriptura de d'un mis, escriu payload d'1 byte
        acall   spi_wr                 ;tramet comanda
        mov     a,misResp               ;hi ha payload com a emissor d'un byte
        acall   spi_wr                 ;tramet dada 000A BCDE
        setb    SBCSN                  ;fi comanda en bus SPI
        setb    TRX_CE                  ;habilita transceiver
envResp1:   jnb     DR,envResp1         ;espera transmissió del preàmbul

```



```

envResp2:    jnb    DR,envResp2    ;espera transmissió del paquet (1 byte)
              clr    TRX_CE        ;deshabilita transceiver
              clr    TX_EN        ;selecciona el transceiver en recepció
resin:        clr    consum        ;actualitza mode baix consum a moderat
              jnb    tramaRF,resin1 ;trama anterior no rebuda, cal resincronitzar
              mov    tempsRF,#10   ;trama anterior rebuda. Inicia tempsRF=10*10 ms (100 ms)
              djnz   resincr,resin5 ;no fi temps resincronitzacio,encara no cal resincronitzar

resin1:       mov    REGX_CTRL,#0x0b ;atura RTC (cal resincronitzar per trama no rebuda)
              setb   tramaRF        ;inicia trama anterior rebuda
              mov    tempsRF,#9     ;resincronització per temps.tempsRF=9*10ms(90+10perduts)
resin2:       mov    a,REGX_CTRL    ;fi temps resincronitzacio,a=REGX_CTRL
              jb     ACC.4,resin2   ;espera fins que REGX_CTRL.4 sigui 0
              mov    b,#14         ;espera uns 0,25*B=3,5ms
              clr    a
resin3:       djnz   ACC,resin3
              djnz   B,resin3
              mov    REGX_MSB,#0
              mov    REGX_LSB,#20-1
              mov    REGX_CTRL,#0x0a ;defineix periodicitat RTC a 10ms i l'enrega
                                      ;nomes cal definir 16 LSB (8 MSB a 0 per defecte)
resin4:       mov    a,REGX_CTRL    ;a=REGX_CTRL
              jb     acc.4,resin4   ;espera fins que REGX_CTRL.4=0
resin5:       pop    psw
              pop    b
              pop    acc
              reti

;subrutina motor
;segons la variable node actualitza el mode de funcionament dels 3 motors del mòdul
;recursos:    a,psw.
;              1 nivell de stack

motor:        clr    estatVib
              mov    a,node
              anl    a,#0x03        ;a=0000 00GH
              jz     mode00
              dec    a
              jz     mode10on
              dec    a
              jz     mode20on
              dec    a
              jz     mode30on
mode00:       clr    P01
              jmp    motor1
mode10on:     djnz   tempsMot+1,fiMotor0 ;no fi temps, no cal fer res
              dec    tempsMot        ;temps(lsb)=0, decrementa temps(msb)

```



```

mov    a,tempsMot          ;a=temps(msb)
inc    a                   ;a=temps(msb) anterior
jnz    fiMotor0            ;no fi temps, no cal fer res mes
jb     P01,mode10off       ;motor funcionant. Ara el parem
mov    tempsMot,#10/256    ;inicia temps en on=10*10ms=100ms
mov    tempsMot+1,#10 MOD 256
setb   P01                 ;motor parat. Ara l'encenem
setb   estatVib
jmp    motor1
mode10off:
clr    P01
mov    tempsMot,#40/256    ;inicia temps en off=40*10 ms=400ms
mov    tempsMot+1,#40 MOD 256
jmp    motor1
mode20on:
djnz   tempsMot+1,fiMotor0 ;no fi temps, no cal fer res
dec    tempsMot            ;temps(lsb)=0, decrementa temps(msb)
mov    a,tempsMot          ;a=temps(msb)
inc    a                   ;a=temps(msb) anterior
jnz    fiMotor0            ;no fi temps, no cal fer res mes
jb     P01,mode20off       ;motor funcionant. Ara el parem
mov    tempsMot,#10/256    ;inicia temps en on=10*10ms=100ms
mov    tempsMot+1,#10 MOD 256
setb   P01                 ;motor parat. Ara l'encenem
setb   estatVib
jmp    motor1
mode20off:
clr    P01
mov    tempsMot,#10/256    ;inicia temps en off=10*10 ms=100ms
mov    tempsMot+1,#10 MOD 256
jmp    motor1
mode30on:
djnz   tempsMot+1,fiMotor0 ;no fi temps, no cal fer res
dec    tempsMot            ;temps(lsb)=0, decrementa temps(msb)
mov    a,tempsMot          ;a=temps(msb)
inc    a                   ;a=temps(msb) anterior
jnz    fiMotor0            ;no fi temps, no cal fer res mes
jb     P01,mode30off       ;motor funcionant. Ara el parem
mov    tempsMot,#2/256     ;inicia temps en on=2*10ms=20ms
mov    tempsMot+1,#2 MOD 256
setb   P01                 ;motor parat. Ara l'encenem
setb   estatVib
jmp    motor1
mode30off:
clr    P01
mov    tempsMot,#2/256     ;inicia temps en off=2*10 ms=20ms
mov    tempsMot+1,#2 MOD 256
fiMotor0:
jmp    motor1
motor1:
mov    a,node+1
anl    a,#0x03             ;a=0000 00GH
jz     mode01

```




```

    dec    a
    jz     mode11on
    dec    a
    jz     mode21on
    dec    a
    jz     mode31on
mode01:   clr    P03
    jmp    motor2
mode11on: djnz   tempsMot+3,fiMotor1    ;no fi temps, no cal fer res
    dec    tempsMot+2                    ;temps(lsb)=0, decrementa temps(msb)
    mov    a,tempsMot+2                  ;a=temps(msb)
    inc    a                             ;a=temps(msb) anterior
    jnz    fiMotor1                      ;no fi temps, no cal fer res mes
    jb     P03,mode11off                  ;motor funcionant. Ara el parem
    mov    tempsMot+2,#10/256             ;inicia temps en on=10*10ms=100ms
    mov    tempsMot+3,#10 MOD 256
    setb   P03                            ;motor parat. Ara l'encenem
    setb   estatVib
    jmp    motor2
mode11off: clr   P03
    mov    tempsMot+2,#40/256             ;inicia temps en off=40*10 ms=400ms
    mov    tempsMot+3,#40 MOD 256
    jmp    motor2
mode21on: djnz   tempsMot+3,fiMotor1    ;no fi temps, no cal fer res
    dec    tempsMot+2                    ;temps(lsb)=0, decrementa temps(msb)
    mov    a,tempsMot+2                  ;a=temps(msb)
    inc    a                             ;a=temps(msb) anterior
    jnz    fiMotor1                      ;no fi temps, no cal fer res mes
    jb     P03,mode21off                  ;motor funcionant. Ara el parem
    mov    tempsMot+2,#10/256             ;inicia temps en on=10*10ms=100ms
    mov    tempsMot+3,#10 MOD 256
    setb   P03                            ;motor parat. Ara l'encenem
    setb   estatVib
    jmp    motor2
mode21off: clr   P03
    mov    tempsMot+2,#10/256             ;inicia temps en off=10*10 ms=100ms
    mov    tempsMot+3,#10 MOD 256
    jmp    motor2
mode31on: djnz   tempsMot+3,fiMotor1    ;no fi temps, no cal fer res
    dec    tempsMot+2                    ;temps(lsb)=0, decrementa temps(msb)
    mov    a,tempsMot+2                  ;a=temps(msb)
    inc    a                             ;a=temps(msb) anterior
    jnz    fiMotor1                      ;no fi temps, no cal fer res mes
    jb     P03,mode31off                  ;motor funcionant. Ara el parem
    mov    tempsMot+2,#2/256              ;inicia temps en on=2*10ms=20ms
    mov    tempsMot+3,#2 MOD 256
    setb   P03                            ;motor parat. Ara l'encenem

```



```

setb    estatVib
jmp     motor2
mode31off: clr    P03
mov     tempsMot+2,#2/256      ;inicia temps en off=2*10 ms=20ms
mov     tempsMot+3,#2 MOD 256
fiMotor1: jmp     motor2
motor2:   mov     a,node+2
anl     a,#0x03                ;a=0000 00GH
jz      mode02
dec     a
jz      mode12on
dec     a
jz      mode22on
dec     a
jz      mode32on
mode02:   clr     P05
ret
mode12on: djnz    tempsMot+5,fiMotor2    ;no fi temps, no cal fer res
dec     tempsMot+4                ;temps(lsb)=0, decrementa temps(msb)
mov     a,tempsMot+4              ;a=temps(msb)
inc     a                        ;a=temps(msb) anterior
jnz     fiMotor2                ;no fi temps, no cal fer res mes
jb      P05,mode12off            ;motor funcionant. Ara el parem
mov     tempsMot+4,#10/256        ;inicia temps en on=10*10ms=100ms
mov     tempsMot+5,#10 MOD 256
setb    P05                      ;motor parat. Ara l'encenem
setb    estatVib
ret
mode12off: clr     P05
mov     tempsMot+4,#40/256        ;inicia temps en off=40*10 ms=400ms
mov     tempsMot+5,#40 MOD 256
ret
mode22on: djnz    tempsMot+5,fiMotor2    ;no fi temps, no cal fer res
dec     tempsMot+4                ;temps(lsb)=0, decrementa temps(msb)
mov     a,tempsMot+4              ;a=temps(msb)
inc     a                        ;a=temps(msb) anterior
jnz     fiMotor2                ;no fi temps, no cal fer res mes
jb      P05,mode22off            ;motor funcionant. Ara el parem
mov     tempsMot+4,#10/256        ;inicia temps en on=10*10ms=100ms
mov     tempsMot+5,#10 MOD 256
setb    P05                      ;motor parat. Ara l'encenem
setb    estatVib
ret
mode22off: clr     P05
mov     tempsMot+4,#10/256        ;inicia temps en off=10*10 ms=100ms
mov     tempsMot+5,#10 MOD 256

```



```

                                ret
mode32on:                     djnz   tempsMot+5,fiMotor2      ;no fi temps, no cal fer res
                                dec    tempsMot+4              ;temps(lsb)=0, decrementa temps(msb)
                                mov    a,tempsMot+4           ;a=temps(msb)
                                inc    a                      ;a=temps(msb) anterior
                                jnz    fiMotor2                ;no fi temps, no cal fer res mes
                                jb     P05,mode32off           ;motor funcionant. Ara el parem
                                mov    tempsMot+4,#2/256       ;inicia temps en on=2*10ms=20ms
                                mov    tempsMot+5,#2 MOD 256
                                setb    P05                    ;motor parat. Ara l'encenem
                                setb    estatVib
                                ret
mode32off:                    clr    P05
                                mov    tempsMot+4,#2/256       ;inicia temps en off=2*10 ms=20ms
                                mov    tempsMot+5,#2 MOD 256
fiMotor2:                     ret

;subrutina initRF
;inicialitza els paràmetres relatius a les transmissions per RF
;recursos:                     psw
;                               1 nivell de stack

initRF:                       clr    SBCSN                    ;comença comanda en bus SPI
                                mov    a,#0x00                ;comanda d'escriptura a partir de registre config 0 (CH_NO)
                                acall   spi_wr                 ;tramet comanda
                                mov    a,#0x7C                ;a=dada per registre 0 (CH_NO)
                                acall   spi_wr                 ;tramet dada (canal 124 => 869,6 MHz)
                                mov    a,#0x0e                ;a=dada per registre 1 (varis)
                                acall   spi_wr                 ;tramet dada (potencia transmissio maxima,
                                                                ;no retransmissio, recepcio normal i 868/915 MHz)
                                mov    a,#0x44                ;a=dada per registre 2 (TX_RX_AFW)
                                acall   spi_wr                 ;tramet dada (adrees de 4 bytes)
                                mov    a,#0x08                ;a=dada per registre 3 (RX_PW)
                                acall   spi_wr                 ;tramet dada (RX payload de 8 bytes)
                                mov    a,#0x01                ;a=dada per registre 4 (TX_PW)
                                acall   spi_wr                 ;tramet dada (TX payload de 1 byte)
                                mov    a,#0xA5                ;a=dada per registre 5 (RX_ADDRESS byte 0)
                                acall   spi_wr                 ;tramet dada
                                mov    a,#0xED                ;a=dada per registre 6 (RX_ADDRESS byte 1)
                                acall   spi_wr                 ;tramet dada
                                mov    a,#0xE6                ;a=dada per registre 7 (RX_ADDRESS byte 2)
                                acall   spi_wr                 ;tramet dada
                                mov    a,#0xDE                ;a=dada per registre 8 (RX_ADDRESS byte 3)
                                acall   spi_wr                 ;tramet dada
                                setb    SBCSN                    ;fi comanda en bus SPI
                                clr    SBCSN
                                mov    a,#0x22                ;a=comanda d'escriptura de TX_ADDRESS des de byte 0

```



```

acall    spi_wr          ;tramet comanda
mov      a,#0xA5        ;a=dada per a byte 0 (TX_ADDRESS byte 0)
acall    spi_wr          ;tramet dada
mov      a,#0xED        ;a=dada per a byte 1 (TX_ADDRESS byte 1)
acall    spi_wr          ;tramet dada
mov      a,#0xE6        ;a=dada per a byte 2 (TX_ADDRESS byte 2)
acall    spi_wr          ;tramet dada
mov      a,#0xDE        ;a=dada per a byte 3 (TX_ADDRESS byte 3)
acall    spi_wr          ;tramet dada
setb     SBSCN           ;fi comanda en bus SPI
ret

```

;subrutina inibateria

;inicialitza el registre ADC-Configuration a fi de monitoritzar el voltatge

;que rep el modul nRF9E5.

;recursos: a,psw

; 2 nivells de stack

```

inibateria: clr      SBSCN          ;comença comanda en bus SPI
mov      a,#0x44          ;comanda d'escriptura register ADC config,el qual son 2
bytes
acall    spi_wr          ;tramet comanda
mov      a,#0x85          ;a=monitoritzacio de VDD/3, PWR_UP=1,VFSEL=0
acall    spi_wr          ;tramet registre de configuracio
mov      a,#0x08          ;a=ADC_RL_JUST=1 justificacio a dreta, DIFFMODE=0,
                          ;RESCTRL=0->6 bits. Així, ADCDATA[5:0]
acall    spi_wr          ;tramet registre de configuracio
setb     SBSCN           ;fi comanda en bus SPI
ret

```

;subrutina lectbateria

;llegeix el V supply (analogic) i guarda a l'acumulador VDD/3 en format digital

;un cop cada 10 segons

;recursos: a,psw.

; 2 nivells de stack

```

lectbateria: djnz     tempsBat+1,fiLect
dec      tempsBat
mov      a,tempsBat
inc      a
jnz      fiLect
acall    inibateria
clr      SBSCN          ;comença comanda en bus SPI
mov      a,#0xc8        ;a=comanda d'inici de conversio ADC de l'entrada VDD/3
acall    spi_wr          ;tramet comanda
setb     SBSCN          ;fi comanda en bus SPI

```



```

fiADC:      jnb     EOC,fiADC           ;espera fins que la conversio sigui completa
            clr     SBSCSN              ;comença comanda en bus SPI
            anl     EXIF,#0xf7          ;esborra interrupcio ADC: IE2=0
            mov     a,#0x40              ;a=comanda de lectura ADC data,començant pel byte 0
            acall   spi_wr              ;tramet comanda
            acall   spi_rd              ;a=ADC DATA byte 0 (byte on es guarda la conversio)
            setb    SBSCSN              ;fi comanda en bus SPI
            mov     bateria,a
            clr     SBSCSN              ;comença comanda en bus SPI
            mov     a,#0x44              ;comanda d'escriptura registre ADC config,el qual son 2 bytes
            acall   spi_wr              ;tramet comanda
            mov     a,#0x81              ;a=monitoritzacio de VDD/3, PWR_UP=0,VFSEL=0
            acall   spi_wr              ;tramet comanda
            mov     a,#0x09
            acall   spi_wr
            setb    SBSCSN              ;fi comanda en bus SPI
            clr     c
            mov     a,bateria
            subb    a,#37d              ;resta 2.1V(aprox)de a=Vsupply amb bit de portar-ne
            mov     node.7,c            ;si Vsupply>=2.1 node.7=0, Vsupply<2.1 node.7=1
            mov     tempsBat,#1000/256  ;inicialització del comptador de temps de bateria
            mov     tempsBat+1,#1000 MOD 256 ;a 1000*10ms=10 s

fiLect:      ret

;subrutina spi_rd
;rep un byte del bus SPI.      par. sortida:      a=byte rebut
;recursos:                    psw
;                               1 nivell de stack

;subrutina spi_wr
;envia un byte pel bus SPI     par. entrada:      a=byte a enviar
;recursos:                    a,psw
;                               1 nivell de stack

spi_rd:
spi_wr:      anl     EXIF,#0xdf          ;esborra interrupcio SPI
            mov     SPI_DATA,a          ;SPI_DATA=dada a enviar (no caldria a spi_rd)
spi_rdwr1:   mov     a,EXIF              ;llegeix SPI_RX (EXIF.5)
            jnb     ACC.5,spi_rdwr1      ;espera fins que SPI_RX=1
            mov     a,SPI_DATA          ;a=SPI_DATA (no caldria a spi_wr)
            ret
            end

```





L. Codi de la llibreria de control RFControlAPI

En aquest apartat s'adjunta el codi elaborat amb el software *Visual Basic 6.0* que gestiona la comunicació entre la unitat central i el computador. Es basa en la utilització d'un controlador del port, d'una llibreria anomenada *MSComm32* que s'instal·la conjuntament amb altres controls *ActiveX* inclosos en el software *Visual Studio*.

La llibreria que s'ha elaborat, anomenada *RFControlAPI* facilita la configuració del port sèrie a fi que es pugui establir comunicació amb el node en diferents versions de Windows®. L'únic requeriment per al funcionament de la llibreria és que es disposi del software *Visual Studio*.

Per a elaborar la llibreria, en primer lloc s'ha definit un objecte del tipus *MSComm* (per aquest motiu és necessària la llibreria que s'ha esmentat).

Seguidament s'han dissenyat 7 rutines públiques (s'hi podrà accedir des de fora de la llibreria) i dues de privades (a les quals només s'hi podrà accedir dins de la pròpia llibreria):

```
Attribute VB_Name = "RFControlAPI"

Option Explicit

Private ComPort As MSComm

Public Function InitCom(nport As Integer) As MSComm

    On Error Resume Next
    Set ComPort = New MSComm

    ' Si el port ja està obert el tanquem
    If ComPort.PortOpen = True Then ComPort.PortOpen = False

    ' Elecció del port
    ComPort.CommPort = nport ' Elecció del port a obrir

    ' Definició de les mides de les cues
    ComPort.InBufferSize = 4096 ' Mida de la cua de recepció
    ComPort.OutBufferSize = 4096 ' Mida de la cua de transmissió

    ' Buidat de les cues
    ComPort.InBufferCount = 0 ' Buidat de la cua de recepció
    ComPort.OutBufferCount = 0 ' Buidat de la cua d'emissió

    ' Paràmetres de la comunicació pel port sèrie
    ' La connexió es realitzarà a 19200 Baudis, amb 8 bits de dades,
    ' 1 bit de stop i paritat parell
    ComPort.Settings = "19200,E,8,1"

    ' Es podrien establir altres paràmetres com el Handshaking (el control de
    ' flux)
    ' Estableix que es llegeixi en format binari (bytes) en comptes de String
```



```

ComPort.InputMode = comInputModeText

' Establiment del nombre de caràcters transmesos abans que es generi
' l'event OnComm
ComPort.SThreshold = 1

' Establiment del nombre de caràcters rebuts abans que es generi
' l'event OnComm
ComPort.RThreshold = 1

' Ara s'obre el port
ComPort.PortOpen = True
If ComPort.PortOpen = False Then
    ' S'ha produït un error en intentar obrir el port sèrie
    Beep
    MsgBox "Error: No s'ha pogut obrir el port COM" & _
        nport, vbOKOnly + vbCritical, "RFCControl"
    Set ComPort = Nothing ' No s'ha pogut obrir el port sèrie
End If
Set InitCom = ComPort ' S'ha obert el port sèrie

```

End Function

```
Private Sub SendChar(Comanda As Byte)
```

```

    ' Envia un byte (un caràcter) pel port sèrie
    Dim Str As Variant
    Str = Chr(Comanda) ' ComPort.Output només envia strings, cal fer una conversió
    ' de la comanda de byte a string per poder efectuar l'enviament per port sèrie
    ComPort.Output = Str

```

End Sub

```
Private Function ReadChar(Value As Byte) As Boolean
```

```

    ' Llegeix els bytes rebuts pel port sèrie. Retorna el valor llegit en format de
    ' byte i retorna un booleà que indica si s'ha pogut efectuar la lectura
    ' Value = 0 si no es rep cap byte pel port, i es retorna FALS

    Dim BinaryData As String
    BinaryData = ComPort.Input

    ' ComPort.Input retorna un string. Fem una conversió de string al valor ascii
    ' de la string i serà el que la funció retornarà (a més del booleà)
    If LenB(BinaryData) > 0 Then
        Value = Asc(BinaryData)
        ReadChar = True
    Else
        ReadChar = False
    End If

```

End Function

```
Public Function EndConnection() As Boolean
```




```

EndConnection = False
If ComPort.PortOpen = True Then
    Dim bTempsExcedit As Boolean, Temps As Long

    ' A partir de l'instant en que es produeixi l'event s'esperarà 5 segons
    ' abans de tancar el port
    bTempsExcedit = False
    Temps = Now
    While ComPort.OutBufferCount > 0

        ' Es permet el processat de missatges pendents
        DoEvents
        If DateDiff("s", Now, Temps) > 5 Or bTempsExcedit = True Then
            Dim resp As VbMsgBoxResult
            resp = MsgBox("Dades no enviades", vbAbortRetryIgnore + vbExclamation, _
                "RFControl")
            If resp = vbRetry Then

                ' Si s'escull tornar-ho a provar es repeteix l'operació
                Temps = Now ' S'ignora el límit de temps
            ElseIf resp = vbAbort Then

                EndConnection = False
                Exit Function
            ElseIf resp = vbIgnore Then
                bTempsExcedit = True
            End If
        End If
    Wend

    ' Ara el buffer de transmissió ja s'ha buidat.
    ComPort.PortOpen = False
End If
EndConnection = True

```

End Function

```

Public Function SetNewState(node As Byte, Mode As Byte) As Byte

```

```

    Dim Value As Byte
    SendChar (2 * Mode + 8 * (node))
    ' En el format dels missatges que espera la unitat central, en cas que sigui un
    ' missatge d'actualització d'estat, el format del missatge esperat és el
    ' següent:
    ' ABCDE FG0, on ABCDE és el codi del node a actualitzar (5 bits de major
    ' significació) i FG és el mode que es vol de vibració.
    ' Així, Byte a enviar = 8*Node+2*Mode

    SetNewState = ReadChar(Value) 'Retorna el valor respost per la unitat central,
    ' que és l'echo de la comanda que la unitat ha rebut per port sèrie i així es
    ' buida el buffer

```

End Function

```

Public Sub ReadNodeBatAndState(node As Byte, battery As Double, state As Byte,
Optional ShowMessage As Boolean = True)

```



```

Dim Value As Byte
Dim HasReadSomething As Boolean
ComPort.InBufferCount = 0 ' buidat de la cua de recepció
SendChar (node * 8 + 5)
Wait 200

' En el format dels missatges que espera la unitat central, en cas que sigui un
' missatge de lectura de bateria i estat, el format del missatge esperat és el
' següent:
' ABCDE 101, a on ABCDE és el codi del node a llegir (5 bits de major
' significació) i 101 és el tipus de missatge. Així, Byte a enviar =8*Node+5
' L'espera de 200ms es fa ja que la unitat central envia un missatge cada
' 100ms. Així ens assegurem que els nodes receptors han tingut temps suficient
' per rebre la comanda.
' A més, si es vol llegir la bateria de diversos nodes, així s'esglaona
' l'enviament de la comanda als diferents nodes (cada 100ms només es llegeix un
' node, així s'eviten col·lisions)

HasReadSomething = ReadChar(Value)
If HasReadSomething = False Then

' Si la unitat central no ha donat una resposta a la comanda és mot probable
' que el node estigui desconnectat de l'alimentació o que sigui inexistent, per
' la qual cosa no respondria a la unitat

If ShowMessage = True Then
MsgBox "No s'ha establert comunicació amb el node", vbExclamation, "Llegir
Bateria"
Else
End If
Else

' La unitat central ha enviat per sèrie la resposta que ha rebut per RF de
' part del node.
' La resposta té el següent format: AB CDEFGH, on AB codifica el mode de
' vibració del node, (són els bits de major pes), i CDEFGH (els 6 bits de menor
' pes) són la conversió de la
' bateria del node de valor analògic a valor digital
' battery = ((Value Mod 64) * 3.66) / 64 'un cop obtingut el residu de la
' resposta dividida per 2^6 (per com s'ha definit el missatge), es té que el
' valor de la bateria serà el real obtingut de (Value Mod 64) * 3.66 / 2^N, on N
' és el nombre de bits que s'usa per a fer la conversió AD en la lectura de la
' bateria del microcontrolador
' state = (Value \ 64) 'fa la divisió entera i ens quedem amb els 2 bits de
' major pes
End If

End Sub

```



```
Public Function SetNewCode(node As Byte) As Byte
```

```
    Dim Value As Byte
    Dim HasReadSomething As Boolean
    ComPort.InBufferCount = 0 'buidat de la cua de recepció
    SendChar (node * 8 + 3)
    Wait 200

    ' En el format dels missatges que espera la unitat central, en cas que sigui un
    ' missatge de nou codi, el format del missatge esperat és el següent:
    ' ABCDE 011, a on ABCDE és el codi del node a emmagatzemar (5 bits de major
    ' significació) i 011 és el tipus de missatge. Així, Byte a enviar =8*Node+3
    ' L'espera de 200ms es fa ja que la unitat central envia un missatge cada
    ' 100ms, així ens assegurem que els nodes receptors han tingut temps suficient
    ' per rebre la comanda.
    ' Cal dir que aquesta operació canvia el valor dels identificadors de TOTS els
    ' nodes que estiguin connectats en aquell moment, per la qual cosa convé tenir
    ' en connexió únicament el node que en el qual vulguem efectuar un canvi de
    ' l'identificador, ja que els identificadors es perdrien

    HasReadSomething = ReadChar(Value)
    If HasReadSomething = False Then
        MsgBox "No s'ha establert comunicació amb el node", vbExclamation, "Nou Codi"
    Else
        SetNewCode = (Value \ 8) 'fa la divisió entera
    End If
```

```
End Function
```



```
Public Function ReadCode() As Byte
```

```

Dim Value As Byte
Dim HasReadSomething As Boolean
ComPort.InBufferCount = 0 'buidat de la cua de recepció
SendChar (1)
Wait 200

' En el format dels missatges que espera la unitat central, en cas que sigui un
' missatge de lectura de codi identificador, el format del missatge esperat és
' el següent:
' XXXXX 001, a on XXXXX és qualsevol codi (5 bits de major significació)
' i 001 és el tipus de missatge. Així, Byte a enviar =1, ja que no ens importen
' els altres bits.
' L'espera de 200ms es fa ja que la unitat central envia un missatge cada
' 100ms, així ens assegurem que els nodes receptors han tingut temps suficient
' per rebre la comanda.
' Cal recordar que per efectuar aquesta operació cal tenir només un node en
' connexió, ja que el que fem és que tots els nodes en funcionament retornin el
' seu codi. Si hi hagués més d'un node en funcionament seria probable una
' col·lisió en els missatges, i no sabriem a qui pertany cada codi

Value = 0
HasReadSomething = False
HasReadSomething = ReadChar(Value)
If HasReadSomething = False Then
    MsgBox "No s'ha establert comunicació amb el node", vbExclamation, "Llegir
Codi"
Else
    ReadCode = (Value \ 8) 'fa la divisió entera, ja que en el format dels
' missatges retornats, els 5 bits de major significació són els que contenen el
' codi identificador del node
End If

End Function
```

```
Public Sub Wait(mSeconds As Integer)
```

```

'Espera els msegons indicats en la variable mSeconds
Dim StartTime
Dim EndTime
Dim Seconds As Double
Seconds = mSeconds / 1000
StartTime = Timer
EndTime = StartTime + Seconds ' Estableix la hora d'inici
Do While Timer < EndTime
Loop

End Sub
```



M. Implementació de la interfície gràfica RFControl

La interfície gràfica serveix per a controlar el sistema a més alt nivell i de manera amigable. De manera que es pretén que el seu funcionament sigui força intuïtiu per a l'usuari.

S'ha elaborat amb el software *Visual Basic* prenent com a base la llibreria de control RFControlAPI, i s'ha estructurat en dues pantalles (formularis principals): la pantalla principal i la pantalla d'estat global de bateria. Totes els altres elements que es visualitzen són únicament missatges i no requereixen de formulari propi, sinó que s'implementen en la mateixa pantalla principal.

M.1. Implementació de la pantalla principal

```
Option Explicit
```

```
Private WithEvents ComPort As MSComm
```

```
Private Sub btnReadBattery_Click()
```

```
    Dim RState As Byte
    Dim RBatt As Double
    Dim NumNode As Byte
    NumNode = Val(NodeIdForBat(0).Text)
    If (NumNode > 0 And NumNode < 32) Then
        ReadNodeBatAndState NumNode, RBatt, RState
        Estat(1).Text = RState
        Bateria(2).Text = RBatt
    Else
        MsgBox "Intoduir un codi entre 1 i 31", vbOKOnly + vbExclamation, "Lectura de
        bateria"
    End If
```

```
End Sub
```

```
Private Sub COM1_Click()
```

```
    COM1.Checked = True
    COM2.Checked = False
    COM3.Checked = False
    ConnectMenu 1
```

```
End Sub
```



```
Private Sub COM2_Click()
```

```
    COM2.Checked = True
    COM1.Checked = False
    COM3.Checked = False
    ConnectMenu 2
```

```
End Sub
```

```
Private Sub COM3_Click()
```

```
    COM3.Checked = True
    COM1.Checked = False
    COM2.Checked = False
    ConnectMenu 3
```

```
End Sub
```

```
Private Sub ConnectMenu(nport As Integer)
```

```
    Set ComPort = InitCom(nport)
    If ComPort Is Nothing Then
        StatusBar.Text = "Desconnectat"
    Else
        StatusBar.Text = "Connectat COM" & nport
        Connect(1).Enabled = False
        Disconnect(2).Enabled = True
        btnRefresh.Enabled = True
        btnReadBattery.Enabled = True
        NewCode(1).Enabled = True
        AllBat(1).Enabled = True
        ReadId(2).Enabled = True
        NodeIdForBat(0).Enabled = True
    End If
End Sub
```

```
Private Sub Disconnect_Click(Index As Integer)
```

```
    Dim resp As Boolean
    StatusBar.Text = "S'està tancant la connexió"
    If ComPort.PortOpen Then
        resp = EndConnection
        If resp Then
            Connect(1).Enabled = True
            Disconnect(2).Enabled = False
            btnRefresh.Enabled = False
            btnReadBattery.Enabled = False
            NewCode(1).Enabled = False
            AllBat(1).Enabled = False
            ReadId(2).Enabled = False
            NodeIdForBat(0).Enabled = False
            StatusBar.Text = "Connexió finalitzada. Desconnectat"
        Else
            End If
        End If
    End If
End Sub
```



```
Private Sub Exit_Click(Index As Integer)
```

```
    Dim resp As Boolean
    If ComPort.PortOpen = True Then
        resp = EndConnection()
        Unload Interficie
    Else
        Unload Interficie
    End If
```

```
End Sub
```

```
Private Sub About_Click(Index As Integer)
```

```
    MsgBox "Programa desenvolupat per Lúdia Alcobero Gabarró" _
        & " per al Projecte de Final de Carrera"
```

```
End Sub
```

```
Private Sub AllBat_Click(Index As Integer)
```

```
    FormBateria.Show vbModal, Me
```

```
End Sub
```

```
Private Sub btnRefresh_Click()
```

```
    Dim element As TextBox
    For Each element In Text1
        If element.Text = "0" Then
            element.BackColor = &HFF00&
        ElseIf element.Text = "1" Then
            element.BackColor = &HFFFF&
        ElseIf element.Text = "2" Then
            element.BackColor = &H80FF&
        ElseIf element.Text = "3" Then
            element.BackColor = &HFF&
        Else
            element.BackColor = &HFF00&
            element.Text = "0"
        End If
        SetNewState element.Index + 1, element.Text
    Next
    MsgBox "Estat del sistema actualitzat", vbInformation, "RFControl"
```

```
End Sub
```



```

Private Sub NewCode_Click(Index As Integer)

Dim Answer As Integer
Dim NodeNou As Byte
Dim NewCode As Byte
Dim EnteredString As String
Dim EnteredValue As String
Dim Finished As Boolean
Answer = MsgBox("Per a efectuar aquesta operació cal tenir un únic node engegat",
vbOKCancel + vbExclamation + vbDefaultButton2, "Atenció")
If Answer = vbOK Then
    Finished = False
    While Finished = False
        EnteredString = InputBox("Introduir nou codi")
        If EnteredString = "" Then
            Finished = True
        Else
            EnteredValue = Val(EnteredString)
            If (EnteredValue > 0 And EnteredValue < 32) Then
                Answer = MsgBox("Desitja sobre escriure el codi de tots els nodes que
estiguin connectats en aquest moment? ", vbOKCancel + vbExclamation, "Atenció")
                If Answer = vbOK Then
                    NewCode = Val(EnteredValue)
                    NodeNou = SetNewCode(NewCode)
                    If NodeNou <> 0 Then
                        MsgBox "El node connectat passa a tenir el codi " & NewCode,
vbOKOnly, "Nou codi"
                    Else
                        End If
                    Finished = True
                End If
            Else
                MsgBox "Intoduir un codi entre 1 i 31", vbOKOnly + vbExclamation, "Nou
codi"
            End If
        End If
    Wend
End If

End Sub

```

```

Private Sub ReadId_Click(Index As Integer)

Dim Ident As Byte
Dim Answer As Integer
Answer = MsgBox("Per a efectuar aquesta operació cal tenir un únic node engegat",
vbOKCancel + vbExclamation + vbDefaultButton2, "Atenció")
If Answer = vbOK Then
    Ident = ReadCode()
    If Ident <> 0 Then
        MsgBox "El codi del node en connexió és " & Ident, vbInformation,
"Lectura d'identificador"
    Else
        End If
    End If
End If

End Sub

```




```
Private Sub Simulate_Click(Index As Integer)
```

```
    StatusBar.Text = "Connectat"
    Connect(1).Enabled = False
    Disconnect(2).Enabled = True
    btnRefresh.Enabled = True
    btnReadBattery.Enabled = True
    NewCode(1).Enabled = True
    AllBat(1).Enabled = True
    ReadId(2).Enabled = True
    NodeIdForBat(0).Enabled = True
```

```
End Sub
```

```
Private Sub Tips_Click(Index As Integer)
```

```
    MsgBox "Si es desitja efectuar una operació de nou codi cal tenir un únic" _
        & Chr(13) + Chr(10) & "node en funcionament. Aquesta operació estableix el  
codi desitjat" _
        & Chr(13) + Chr(10) & "a TOTS els nodes que estiguin en connexió en aquell  
moment." _
        & Chr(13) + Chr(10) & " " _
        & Chr(13) + Chr(10) & "Si es necessita llegir el codi d'un node també cal  
que aquest sigui" _
        & Chr(13) + Chr(10) & "l'únic en funcionament a fi de saber quin és el node  
que ha respost" _
        & Chr(13) + Chr(10) & "i per evitar col·lisions" _
        & Chr(13) + Chr(10) & " " _
        & Chr(13) + Chr(10) & "Si es vol llegir l'estat de tots els nodes és  
imprescindible esperar que" _
        & Chr(13) + Chr(10) & "l'operació s'hagi completat abans d'efectuar cap  
altra operació, doncs" _
        & Chr(13) + Chr(10) & "tarda uns segons a finalitzar" _
        & Chr(13) + Chr(10) & " " _
        & Chr(13) + Chr(10) & "L'estat de vibració dels nodes es codifica de la  
següent manera: " _
        & Chr(13) + Chr(10) & " " _
        & Chr(13) + Chr(10) & "Mode 0: motor aturat" _
        & Chr(13) + Chr(10) & "Mode 1: el motor vibra durant 100ms cada 500ms (D =  
20%, f = 2 Hz)" _
        & Chr(13) + Chr(10) & "Mode 2: el motor vibra durant 100ms cada 200ms (D =  
50%, f = 5 Hz)" _
        & Chr(13) + Chr(10) & "Mode 3: el motor vibra durant 20ms cada 40ms (D  
= 50%, f =25 Hz)", vbInformation, "Recomanacions bàsiques per a l'ús de  
RFControl"
```

```
End Sub
```





M.2. Implementació de la pantalla d'estat global de bateria

```
Private Sub btnRefresh_Click()  
  
    Dim element As TextBox  
    Dim node As Byte  
    Dim battery As Double  
    Dim state As Byte  
  
    FormBateria.MousePointer = vbHourglass  
    btnRefresh.Enabled = False ' desactivem l'opció de llegir per evitar  
                                ' lectures innecessàries  
    For Each element In Text1  
        StatusBar.Text = "Llegint la bateria del node" & (element.Index + 1)  
        battery = 0  
  
        ReadNodeBatAndState element.Index + 1, battery, state, False  
        element.Text = battery  
  
        If battery < 2.1 Then  
            If battery <> 0 Then  
                element.BackColor = &HFF&          ' vermell  
            ElseIf battery = 0 Then  
                element.BackColor = &H8000000F      ' blanc  
            End If  
  
            Else  
                element.BackColor = &HFF00&        ' verd  
            End If  
  
        End If  
  
    Next  
  
    StatusBar.Text = "Comanda completada"  
    FormBateria.MousePointer = vbDefault  
    MsgBox "Estat de bateria del sistema actualitzat", vbInformation, "RFControl"  
  
End Sub
```



